

Nº9 – Septiembre 2004

al descubierto **LIVINGSTONE, SUPONGO**

CREANDO UNA AVENTURA CONVERSACIONAL CON Z88DK (y III)

ENTREVISTA A DROY input

zona www **NAME THE GAME**

...y mucho más

www.speccy.org/magazinezx



Año II Nº 9 Septiembre 2004





Redacción:

Santiago Romero (SROMERO).

Federico Álvarez (FALVAREZ).

Pablo Suau (SIEW).

Miguel A. García Prada (DEVIL_NET).

Ilustración de Portada:

Juanje Gómez (DEV).

Colaboraciones en este número:

Josetxu Malanda (HORACE).

SpeccyTourPlayer

Maquetación en PDF

Álvaro Alea (ALEASOFT)

Contacto:

magazine@speccy.org



Redacción de MAGAZINE ZX

Ya estamos aquí de vuelta tras el caluroso mes de agosto, y esperando que todos vosotros hayáis podido disfrutar de unos merecidos días de descanso. Lentamente, todo vuelve a la normalidad y la rutina trata de apoderarse una vez más de nuestras vidas. Para tratar de evitarlo, os presentamos este nuevo número de vuestra revista Magazine ZX.

En esta entrega tras el lapso estival los contenidos no son demasiado extensos, si bien esperamos que sean de vuestro agrado. Tras el consabido repaso a la actualidad en lo referente al Spectrum, analizamos tres juegos no demasiado conocidos, de la mano de FALVAREZ. Tras abrir boca, NÉSTOR LUCAS pone patas arriba todo un clásico como es Livingstone Supongo.

SIEW sigue desentrañando los secretos de la herramienta z88dk. En esta entrega completaremos la aventura conversacional de Guybrush Threepwood, aprendiendo nuevas funcionalidades que nos serán de gran utilidad a la hora de elaborar nuestros propios programas.

En la parte más subjetiva de la publicación, analizamos la web Name The Game, un soplo de aire fresco en el panorama de los *sites* sobre Spectrum. Para cerrar este número, contamos con la entrevista que Droy concedió a nuestro colaborador HORACE y la opinión de un jugón del Speccy Tour acerca del devenir de dicha competición, cuya edición de este año está apunto de comenzar.



Redacción de MAGAZINE ZX

Ya ha pasado el verano, algunos se han llevado el Spectrum a la playa, otros a la montaña, otros dentro de una GBA o un teléfono móvil, y otros le han dado al emulador. Pero, ¿qué ha pasado durante estos días de descanso? Demos un repaso a la actualidad:

Nuevas versiones de emuladores

En este intervalo de tiempo algunos emuladores se han actualizado, como es el caso de SpeccyalK (para Dreamcast y GP32), SpeXtrum (emulador para XBox), Q-Emulator (emulador de QL para Windows y Mac OS), ZXSpin (versión 0.504) y Real Spectrum (R14), siendo ésta la última la última versión para MS-DOS. Parece ser que Ramsoft va a centrar sus esfuerzos en la versión para Windows. ¿Veremos alguna vez la versión Linux?

Quedadas y parties

Durante este verano se han celebrado importantes reuniones de usuarios, como han podido ser la Campus Party o la Euskal Party. En concreto, en esta última hubo una sección específica dedicada a la retroinformática, en la que tuvo lugar una interesante conferencia en la que participaron como ponentes gente de la talla de S.T.A.R., Rafa Corrales, Injaki o nuestro colaborador Horace, y una competición de juegos organizada por Iván Ruiz.

También se celebró en Torrente (Valencia), organizada por usuarios de Atari, la party Vintagenarios, en la que se reunieron la flor y nata de los sistemas *vintage*. Esperamos una crónica detallada de lo que aconteció en la web de Matranet.

Suspensión ¿temporal? de La Tira de speccy.org

Se acabó: este es el título de la última tira dibujada por Pedrete. Esa tira que semana tras semana nos comentaba la actualidad en el mundillo y en las news en clave de humor. Tal como explica Pedrete en la propia página de la tira, durante el mes de Septiembre se van a producir unos cambios importantes en su vida personal que van a hacer imposible, al menos de momento, que continúe trabajando en ella. Sólo nos queda agradecer sinceramente a Pedrete el trabajo realizado y desearle mucha suerte en sus nuevas circunstancias. Nosotros estaremos aquí esperando a que vuelva, releyendo las tiras anteriores que seguirán online en el servidor de speccy.org. Gracias, Pedrete, y suerte.

El Speccy Tour 2004 calienta motores

Primeras noticias del nuevo tour: como siempre, Iván Ruiz nos mantiene informados en es.comp.sistemas.sinclair. Entre las novedades, un control automatizado de la competición basado en PHP, gracias a la colaboración de Kak, lo que hará que no se dependa tanto del correo electrónico durante el desarrollo de la competición. Se supone que esto va a tener como consecuencia la disminución de los retrasos al hacerse la gestión del tour mucho más sencilla.

Los emuladores que se podrán usar serán Real Spectrum en sus versiones DOS y WIN32, FUSE y Spectaculator 5 (si el autor del programa no pone ningún inconveniente). SPIN queda fuera al negarse los autores a incluir un método de competición en su emulador (la cual es su decisión y es totalmente respetable). Mientras esperamos nuevas noticias, seguiremos entrenando para mantener los buenos

resultados españoles en la competición. Que lástima que el Speccy Tour no sea categoría olímpica, porque si no seguro que hubiéramos acabado en un puesto mucho más elevado en el medallero gracias a nuestros jugones.

En principio, parece que la inscripción ya estará abierta en el momento de publicar este número de la revista, por tanto, os animamos desde aquí a todos a que participéis. Es una experiencia única, y una excusa sin igual para jugar y jugar a nuestro querido Spectrum, además de servir posiblemente para descubrir nuevos juegos o mejorar nuestro nivel en algunos ya conocidos.

Froggermix, nuevo juego español en desarrollo para ZX Spectrum

A finales del mes de julio, Jaime Tejedor (metalbrain) sorprendía a todos anunciando el comienzo de un nuevo proyecto, ideado por José Juan Ródenas (sejuan) y con él mismo y Juan Carlos Soriano (JC) como colaboradores, cuyo objetivo es la creación de un nuevo juego para Spectrum. De momento sólo encontraremos en la página del proyecto algunas ideas, el argumento, y algunos estupendos gráficos del juego que se pretende desarrollar (Frogger Mix).

Cuando se anunció el proyecto eran esas tres las personas que de momento estaban subidas al carro, y dicho anuncio se realizó con la intención de que la gente pueda participar. De hecho, según el mismo Metalbrain asegura, se trata de un desarrollo abierto, cuyo código sería publicado en la lista de correo z80asm. Sin embargo, desde dicho anuncio no se ha vuelto a saber nada del tema, y la página, diseñada por nuestro redactor MIGUEL sigue tal cual estaba entonces. Esperamos que tras la vuelta de vacaciones se reanude el proyecto y se reinicie la larga tradición de producción de software para nuestra entrañable máquina dentro de nuestras fronteras.

La escena rusa sigue en la brecha

Novedades software: allá en la helada Rusia la escena sigue muy activa. Prueba de ello son las nuevas demos y programas que se anuncian año tras año. Como siempre, entre las demos las hay de muy alto nivel, exprimiendo la máquina al máximo y demostrando un gran conocimiento del hardware. Se han lanzado también dos juegos nuevos, uno de ajedrez y una aventura Hentai (aventura erótica manga) que, a menos que sepamos ruso, por desgracia no vamos a poder disfrutar.✍

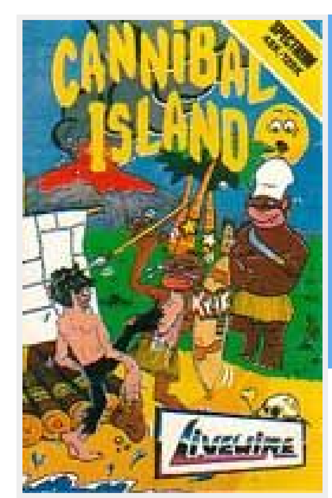
LINKS

- SpeccyalK: http://www.speccyal.be/
- ZXSpin: http://www.zxspin.co.uk/
- Q-Emulator: http://users.infoconex.com/~daniele/q-emulator.html
- Real Spectrum: http://www.ramsoft.bbk.org/realspec.html
- Conferencia Retroeuskal 2004 (renombrar extensión .mp3 a .spx): http://www.euskadigital.net/enredando_net/euskal_12/audio/12.mp3
- Vintagenarios: http://www.matranet.net/PARTY/
- La Tira de speccy.org: http://www.speccy.org/pedrete/
- Speccy Tour 2004: http://www.speccy.org/st04/
- Froggermix: http://www.speccy.org/froggermix/
- Chaos Constructions 2004 (scene rusa): http://cc.enlight.ru/



En esta ocasión, hemos seleccionado tres títulos no muy conocidos para proceder a su análisis. Son los siguientes: Cannibal Island, Stack Up y The Island Of Dr. Destructo. Esperamos que sean de vuestro agrado.

CANNIBAL ISLAND



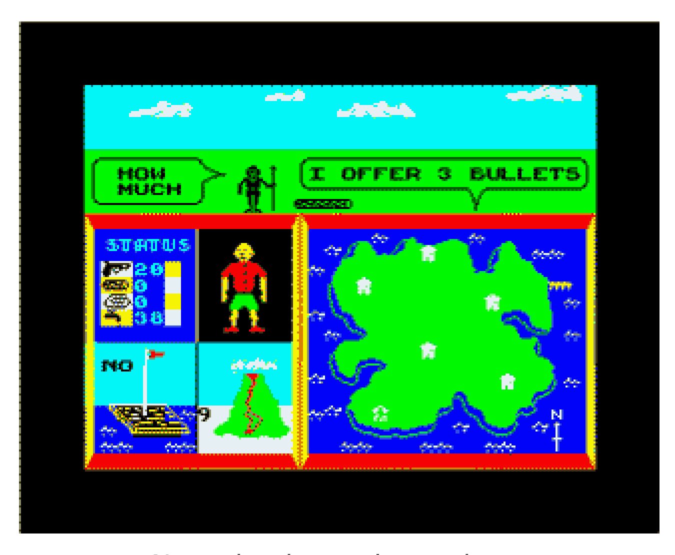
Título	Cannibal Island
Género	Estrategia
Año	1986
Máquina	48K
Jugadores	1 Jugador
Compañía	Live-Wire Software
Autor	Barry Jones
Otros comentarios	

- N/A

Eres el único superviviente de un naufragio, y has ido a parar a una isla habitada por caníbales. Debes escapar, pero además el tiempo corre en tu contra, ya que en la isla hay ¡¡¡un volcán a punto de estallar!!!

La pantalla de juego

Este es el argumento que nos ofrece esta creación de Barry Jones. Para escapar, deberemos construir una balsa, intercambiando con los nativos nuestras balas por troncos y cuerdas. Tendremos que estar atentos a lo que encontremos, ya que si vemos huesos, posiblemente los caníbales anden por allí cerca. Podremos dispararlos, pero teniendo en cuenta que las balas son un bien escaso. Pero si no nos defendemos y los caníbales nos atacan, perderemos un miembro de nuestro cuerpo.

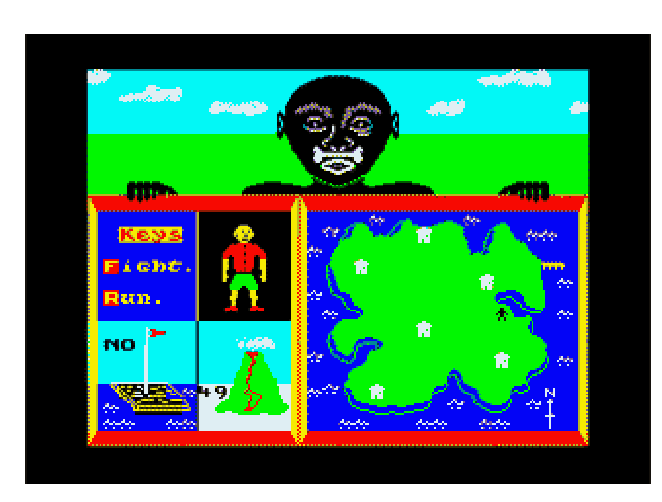


Negociando con los nativos

A esta tragedia se añade el caluroso clima, que nos obligará a aprovisionarnos de agua más a menudo de lo que desearíamos. Así que, cuando hayamos conseguido todas las piezas para construir la balsa, debemos encaminarnos con presteza al embarcadero que se encuentra al noreste de la isla.

Para empezar, el juego consta de 3 niveles de dificultad, de forma que podamos ajustarla a nuestra pericia. Una vez comenzada la partida, pulsando la tecla i, el programa nos muestra las teclas con las que manejaremos a nuestro protagonista, que nos permitirán movernos por la isla, disparar en 4 direcciones, negociar con los nativos y, una opción nada común, como es grabar nuestros progresos en un microdrive.

El desarrollo del juego es tal y como lo hemos comentado más arriba. Recorreremos la isla en busca de poblados donde podemos negociar con los nativos para intercambiar troncos y cuerdas por nuestras balas, y así poder construir la balsa. Pero los nativos no son tontos, si rechazan nuestra oferta tendremos que intentar hacer el trueque con más balas.

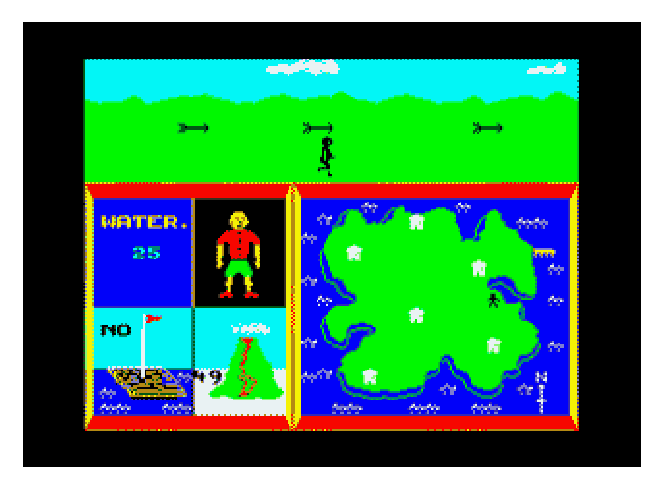


¡Peligro! Los caníbales nos ataca

Fuera de los poblados encontraremos pozos de agua, donde rellenar nuestra cantimplora, y alguna bala que otra tirada en el suelo, que nos vendrá muy bien a la hora de negociar. Pero, por desgracia, la isla está plagada de caníbales. Cuando nos encontremos con uno de ellos tendremos la posibilidad de luchar o salir huyendo. Para salir por piernas debemos contar con agua suficiente, ya que esquivar las flechas es muy cansado, y si no reponemos bebiendo desfalleceremos a causa de la sed y seremos cocinados por los caníbales. Si nos enfrentamos a ellos, se produce un combate por turnos donde aleatoriamente el caníbal o nosotros iremos sufriendo heridas. Al superar las 10 heridas, el perdedor morirá, y el ganador seguirá su camino.

Por tanto es buena idea intentar memorizar dónde están los pozos, para estar siempre dispuestos a escapar corriendo, ya que el resultado de los combates es totalmente aleatorio.

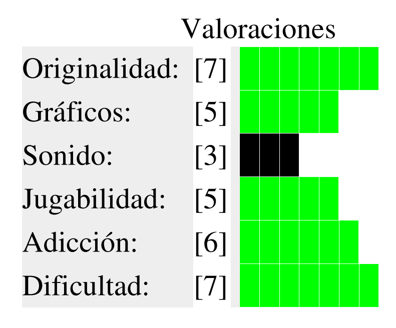
Técnicamente, el juego es muy simple y no demuestra nada. A veces tarda en responder a la pulsación de las teclas. Gráficamente tampoco es nada del otro mundo. El sonido es lamentable, sólo algunos pitidos a modo de efectos al movernos o realizar acciones.



Escapando de las flechas de los caníbales

En resumen, un título con una buena idea, sencillo, pero flojo técnicamente, que nos puede llegar a enganchar como mucho una tarde intentando escapar de la isla, lo cual no es demasiado fácil, pero que pierde interés enseguida debido a lo repetitivo de su desarrollo y a la ausencia de alardes técnicos en su programación.

Valoraciones

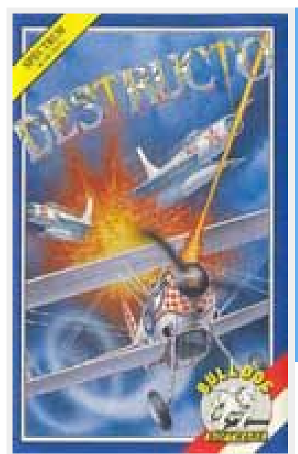


Originalidad:	[7]	
Gráficos:	[5]	
Sonido:	[3]	
Jugabilidad:	[5]	
Adicción:	[6]	
Dificultad:	[7]	

Descárgalo de:

- WOS
  http://www.worldofspectrum.org/infoseek.cgi?regexp=Cannibal+Island

THE ISLAND OF DR. DESTRUCTO



Título	The Island Of Dr. Destructo
Género	Arcade - Shoot'em Up
Año	1987
Máquina	48K
Jugadores	1 ó 2 Jugadores
Compañía	Bulldog
Autor	Eugene Messina, David Lincol-Howells
Otros comentarios	

- Crash Issue 44
- Your Sinclair Issue 22

Este Dr. Destructo se trata de un *mata-mata* de serie barata (o budget). En estas ocasión la historia es la siguiente: Tenemos 7 días para abortar los plantes de dominación global del Dr. Destructo. Para evitarlo, nuestra misión consiste en destruir el imperio del Dr. Destructo. El juego consta de 21 pantallas que se desarrollan durante días y noches a medida que vamos progresando en la historia.



El menú de opciones

El juego consiste en ir destruyendo la flota de nuestro enemigo a base de lanzar sobre ella los aviones que vayamos derribando en el cielo. En cada pantalla, además, contaremos con una bomba que podremos lanzar en el lugar que prefiramos. Una vez destruido el objetivo, pasaremos a la siguiente pantalla. Hay que destacar que pueden participar dos jugadores cooperativamente, con lo que el juego quizás gane en competitividad, para ver quién de los dos consigue una mayor cantidad de puntos.

Podemos controlar nuestro avión con el teclado o con cualquier joystick de los habituales (kempston, sinclair o cursor). Las aeronaves obedecen rápidamente y con precisión a nuestras órdenes. El sonido es correcto, con efectos para los disparos y explosiones, y poco más.

Se trata, en resumidas cuentas, del típico juego barato (estamos hablando de un precio de 1,99 libras), sin más pretensión que entretenernos un rato. Quizá en su momento tuvieran validez este tipo de iniciativas, pero en la actualidad, con el acceso que tenemos a una gran cantidad de títulos, juegos como éste pasen desapercibidos, con bastante justicia por otra parte.



En el fragor de la batalla

Valoraciones

Originalidad:	[5]
Gráficos:	[5]
Sonido:	[6]
Jugabilidad:	[6]
Adicción:	[4]
Dificultad:	[6]

Trucos:

Puedes encontrarlos en The Tip Shop http://www.the-tipshop.co.uk/cgi-bin/search.pl?name=Dr+Destructo

Descárgalo de:

- WOS http://www.worldofspectrum.org/infoseek.cgi?regexp=^Dr.+Destructo$&pub=^Bulldog$

STACK UP



Título	Stack Up
Género	Puzzle
Año	1991
Máquina	48K
Jugadores	1 Jugador
Compañía	Zeppelin Games Ltd
Autor	David Sanders
Otros comentarios	

- Crash Issue 90
- Your Sinclair Issue 68

Estamos ante otro juego de puzzles que destaca por su sencillez y su adicción. En este caso la tarea es sencilla. Se trata de una versión más del tan manido juego de las tres en raya. El juego consiste en ir colocando las piezas que caen de tres en tres desde la parte superior de la pantalla de modo que queden alineadas tres piezas del mismo color, ya sera en horizontal, vertical o diagonal. Cuando esto ocurra, las piezas alineadas desaparecerán y el resto se recolocarán dejándose caer hacia los huecos que se han abierto.

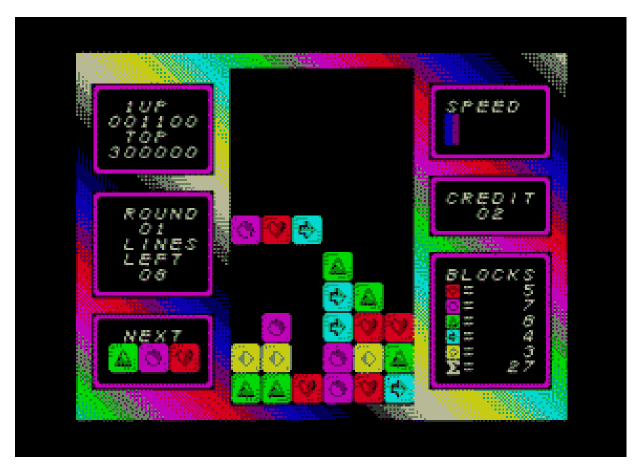
Conforme vamos alineando piezas, el nivel de dificultad irá en aumento. Concretamente, la velocidad con la que caen las piezas crecerá, como ya es típico en este tipo de juegos.



El menú principal

Tampoco vamos a comentar mucho más acerca de la mecánica del juego. Todo dependerá de nuestra pericia a la hora de colocar los bloques que vayan apareciendo y que los colores de los mismos se adecúen a nuestros intereses. Por si queremos ponernos las pilas, tenemos en el menú una opción para practicar antes de afrontar el reto, eligiendo nosotros la velocidad a la que se desarrollará el juego.

Eso sí, técnicamente es bueno. Gráficamente es un juego muy vistoso, y los bloques van descendiendo suavemente pixel a pixel, pero sin merma alguna de velocidad. El uso del color es acertadísimo. La respuesta a los controles (teclado o joysticks kempston, sinclair o cursor) es inmediata. Las melodías (en el menú y durante el juego) están bien aunque, con el tiempo, esta última puede llegar a resultar un poco machacona.

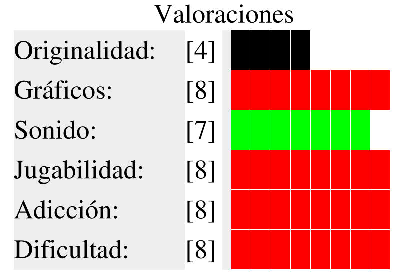


Tratando de conseguir 3 en raya (o más)

A modo de valoración final, se trata de un título correcto, muy bien resuelto técnicamente y adictivo, que nos hará pasar ratos entretenidos si somos amantes de este tipo de juegos.

Valoraciones

Originalidad:	[4]	
Gráficos:	[8]	
Sonido:	[7]	
Jugabilidad:	[8]	
Adicción:	[8]	
Dificultad:	[8]	



Trucos:

Puedes encontrarlos en The Tip Shop http://www.the-tipshop.co.uk/cgi-bin/search.pl?name=Stack+Up

Descárgalo de:

- WOS http://www.worldofspectrum.org/infoseek.cgi?regexp=^Stack+Up$

FALVAREZ



NÉSTOR LUCAS

LIVINGSTONE SUPONGO

Sin duda alguna, una de las compañías españolas de software de entretenimiento de la edad dorada que más nos ha hecho disfrutar, junto con Dinamic, fue sin duda Opera Soft. Bajo este sello se publicaría en 1988 la genial "Abadía del Crimen". Pero no va a ser este el juego que ocupe nuestro patas arriba.



Título	Livingstone Supongo
Género	Arcade/Plataformas
Año	1987
Máquina	ZX Spectrum 48K/128K
Jugadores	1 Jugador
Compañía	Opera Soft

Los chicos de Opera publicaron en 1987 dos juegos de plataformas con los que demostraron su buen hacer para con cuantas plataformas disponibles había en el mercado. Me refiero a "Goody" y a "Livingstone Supongo", siendo este último al que dedicamos este artículo.

Tengo que reconocer que una de las cosas que más me llamó la atención de estos juegos en 1987 fue verlos funcionar en un PC, en la sección de informática de cierta cadena de centros comerciales. Uno se quedaba embobado viendo como se vendía la gran potencia de los PCs de la época con estos juegos, y luego comprobaba que en nuestro querido Speccy era igual, y hasta añadía colores.

Dos años después, en 1989, y tras la sorpresa del final de la primera parte, Opera decidió publicar "Livingstone Supongo 2", juego con poco éxito, en el que se sacrificó (para mi gusto) la jugabilidad por aportar personajes más grandes.

Este patas arriba, como ya es cosa habitual, va acompañado de una partida grabada en el formato RZX con la resolución del juego. Y como novedad, dado que no lo encontre por ningún sitio, he realizado un mapa del juego, que también va incluido en el paquete.

Recordaros además que aprovechéis las posibilidades de grabar posiciones de partida para avanzar en el juego. En la guía paso a paso de la resolución os he indicado tres puntos negros de la partida en la que es más que recomendable recurrir a esta grabación.

Y ahora sí, sin enrollarme más, pasamos al patas arriba propiamente dicho.

PONIÉNDOLO TODO PATAS ARRIBA

Recordemos de forma rápida los controles del juego: para variar, con teclado, las tradicionales teclas opqa, agachándonos con la tecla 'a', y saltando con la tecla 'q'. Disponemos además de cuatro objetos: el boomerang, la espada, la granada y la pértiga. Para usar cualquiera de ellos hace falta seleccionarlos primero: el '1' para el boomerang, el '2' para la espada, el '3' para la granada y el '4' para la pértiga. El disparo es el espacio, y en un momento explicaré como utilizarlo.

En la pantalla encontraremos unos indicadores en la parte inferior izquierda. Son unas barras de progreso, tres para ser exactos, aunque nada más empezar solo veremos dos. Estas indican las reservas de alimentos y de agua que llevamos, siendo necesario reponerlas cada cierto tiempo.

La tercera barra/indicador muestra la fuerza que aplicamos al objeto que tengamos seleccionado en ese momento. Esta fuerza aumenta en función del tiempo que mantengamos pulsada la tecla de disparo (recordemos, la barra espaciadora).

Para practicar un poco como se maneja esto de los disparos, podemos hacerlo al comenzar el juego, sin seleccionar ningún objeto. Veremos como nuestro intrépido aventurero lanza un puñetazo al aire (algo que seamos serios, no sirve de mucho

allá donde nos adentramos).

Al lado del marcador vemos una serie de sombreros. Cada uno de ellos representa una vida. Empezamos el juego con siete vidas, lo que aunque parezca lo contrario, no suele ser suficiente hasta que hemos aprendido bien todos los entresijos de este juego.

Como nota final, decir que cada 20000 (sí, veinte mil) puntos, obtenemos una nueva vida, al igual que al recoger cada una de las cinco joyas.

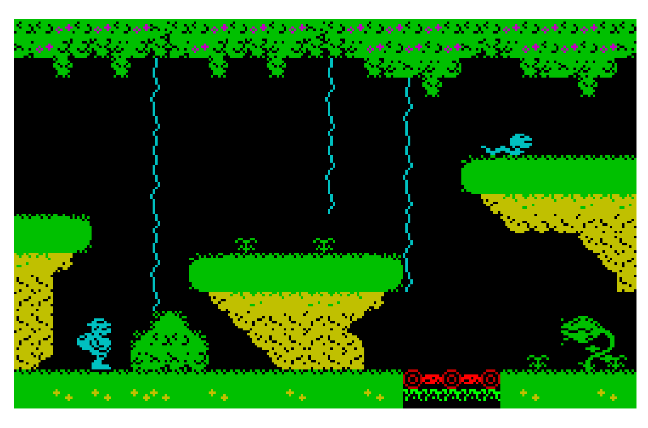
COMIENZA EL JUEGO

Nuestro objetivo en una primera fase del juego es recoger una serie de joyas que se encuentran desperdigadas por la selva. Son cinco, y si nos falta alguna de ellas, el guardian del templo que da acceso a donde se supone que se encuentra Livingstone no nos dejará pasar. Volveremos entonces por arte de magia a uno de los puntos donde nos falte una joya por recoger, y tendremos que atravesar de nuevo la selva llena de peligros.

En esta primera pantalla vemos una serpiente en una loma a la derecha. Lo mejor se subirse a la loma central, y tratar de eliminarla con un boomerang o con una granada, eso lo dejo a vuestro gusto. Eso sí, cuidado con la granada. Si nos cae cerca nos quita una vida. Y del boomerang, no os preocupéis en recogerlo. Digamos que es como si tuvieramos un poke de munición infinita, pero que en realidad no es poke.

Antes de pasar a la siguiente pantalla, hacia la derecha, conviene resaltar un par de puntos. En la parte inferior vemos una planta que se mueve. Es, efectivamente, una planta carnívora. Si nos acercamos mucho a ella, nos quita un punto de vida.

Al lado de la planta carnívora hay una cosa roja en el suelo. Es en realidad una trampa. Si caemos en ella, tendremos que ingeniárnoslas para salir, activando la palanca que abre la puerta que nos retorna a la libertad.



Comienza el juego

Bueno, una vez eliminada la serpiente, nos subimos a la loma donde se encontraba, y seguimos caminando hacia la derecha, entrando en la segunda pantalla.

En esta pantalla tenemos de nuevo tres lomas, y una trampa en el suelo. Además, entre la jarra de agua y la barra de pan que hay tras la segunda loma vemos una zona de arenas movedizas. Y en la loma de la derecha, de nuevo nos encontramos con una serpiente.

La mejor manera de matar a esa serpiente es con la espada, desde la primera loma, cargando la fuerza al máximo. Saltamos de loma a loma hasta la de la derecha, y pasamos a la tercera pantalla.



Allá donde se encuentre una palmera, habrá un mono tirándonos cocos

En la tercera pantalla hay de nuevo una planta carnívora, una serpiente y arenas movedizas. El truco aquí está en colocarse al borde de la loma, y dejar caer una granada cuando este la serpiente debajo. Cuando hayamos conseguido eliminar a la serpiente, nos dejamos caer (no hay peligro), y nos separamos de la planta carnívora la distancia de una pértiga. Nos olvidamos por ahora de las provisiones que había en la pantalla anterior, miramos a la derecha, seleccionamos la pértiga, y la usamos con una fuerza de unos seis puntos. De esta forma deberemos caer en la loma de la derecha, sin pasar a la siguiente pantalla.

Antes de avanzar, las precauciones. En la siguiente pantalla tendremos a dos monos sobre sendas palmeras lanzándonos cocos, fáciles de eliminar con un boomerang a fuerza cinco. A nivel de suelo, justo a la altura del borde de la loma, hay una trampa. Pero lo que conviene tener en cuenta en esta pantalla es que es al aire libre. Y esto significa que aparecerá un aguilucho tratando de atraparnos y llevarnos a su nido. Por ahora vamos a esquivarle todo lo que podamos (esto incluye matarle si es preciso).

Bueno, pues ya sabiendo lo anterior, seleccionamos el boomerang, entramos, y nada más entrar nos agachamos (así los cocos no pueden alcanzarnos). Disparamos el boomerang (recordemos, a cinco de fuerza), y veremos como elimina a ambos monos. Si sale el aguilucho, podemos optar por matarle, o por regresar a la pantalla anterior y volver a empezar.

Una vez eliminados los monos, saltamos hacia adelante desde el borde de la loma, cogiendo velocidad antes, para caer pasada la primera palmera. Saltamos a la siguiente loma (con pértiga o sin ella, se puede de las dos formas).

En la siguiente pantalla tenemos un mono sobre una palmera lanzando cocos (fácilmente eliminable con una granada a máxima fuerza). Hay también un pez, que si bien podemos pasar sin matarle, conviene eliminarle para disminuir riesgos (de nuevo la granada), y con cuidado de que no nos pille el aguilucho. Subimos al tronco que hay sobre el agua en el extremo izquierdo, y seleccionamos el boomerang, cargándolo a máxima fuerza.



La pantalla decisiva del juego. Primero hay que ir donde está el mono, para regresar después y colarse por la gruta debajo de la cascada

Una vez aparezcamos en la siguiente pantalla, lanzamos el boomerang, eliminando al mono que hay en la parte derecha. Seleccionamos la pértiga, nos ponemos en el extremo izquierdo del tronco, y cargamos la pértiga al máximo, saltando (usándola) justo un cuando el tronco deja la cascada. De esta forma aterrizaremos sobre la loma, en la siguiente pantalla.

Sin parar, seguimos caminando hasta caer por el borde, y caer entre los dos indígenas que nos disparan dardos y flechas. Nos agachamos (así no nos pueden dar), y seleccionamos el arma de nuestra opción (aquí va bien tanto la espada como el boomerang). Conviene eliminar primero al que esta a nuestra izquierda, con la cerbatana. Una vez eliminados ambos, seleccionamos la pértiga, y nos situamos a una distancia de una pértiga del banco de arenas movedizas. Cargamos la pértiga a un 8.75 de fuerza, procurando no llegar a 9, siempre por encima de 8.

Si lo hacemos bien, pasaremos casi de largo por encima de un poblado indígena. En realidad, caeremos sobre la casa más alta, justo recogiendo una de las cinco joyas que nos hacen falta para que el guardián del templo nos deje pasar. De la torre alta saltamos a la de la derecha, caminando hacia la derecha, hacia la siguiente pantalla, sin parar en ningún momento (hay que evitar los dardos).

Regresamos a la pantalla anterior, ahora a nivel del suelo. Nada más entrar, nos agachamos. Seleccionamos el boomerang, y lo lanzamos con una fuerza de 5, para eliminar al brujo que hay sobre esa especie de vasija gigante. Una vez eleminado, saltamos por encima de la trampa, situándonos más o menos en el centro de la pantalla. Con el boomerang o con las granadas, a nuestro gusto, eliminamos a los indígenas de las torres de media altura. Podemos olvidarnos del de la torre alta.

Aproximadamente bajo este último indigena es donde debemos situarnos ahora. Mirando hacia la derecha, seleccionamos la pértiga y la usamos con fuerza 7.5. Si nos hemos situado bien, pasaremos de largo la pantalla siguiente, aterrizando en la que va a continuación.

En esta nueva pantalla tenemos tres indígenas, de los cuales, por ahora, solo uno puede alcanzarnos con sus lanzas. Nos situamos debajo de la puerta de la cabaña que hay en el alto, con la granada seleccionada, y al máximo de fuerza. La lanzamos, y nos quitaremos de un plumazo al indígena de la cabaña y al de la torre alta.

En ese momento debería hacer su entrada por la izquierda un soldado. Es enemigo, y va a dispararnos. Y como viene siendo habitual, agacharse es la mejor manera de evitar que nos alcance con su armamento. Lo podemos eliminar con un boomerang o con una espada, lo que más rabia nos dé.

Ahora solo nos queda el indígena de la vasija, que como los anteriores, lo podemos eliminar más rápido con una granada.

Con el camino despejado, saltamos sobre la vasija (podemos hacerlo sin la pértiga), caemos a la derecha recogiendo los víveres (lo que tiene forma de hamburguesa), saltamos por encima de la trampa, y pasamos a la siguiente pantalla.

En esta pantalla es recomedable permanecer agachado nada más entrar. Seleccionamos el boomerang, y con una fuerza de unos 4.8 a 5.2, lanzarlo. De esta forma nos quitamos de enmedio al indígena que hay en la antes de la vasija, al brujo que hay sobre la misma, y si conseguimos una fuerza exacta de 4.8, al que tenemos en la plataforma superior. En cualquier caso, este último indígena no es peligroso. Sí lo es en cambio el que se encuentra en la parte derecha de la pantalla, con la cerbatana, motivo por el que debemos seguir agachados. De nuevo con el boomerang, esta vez al máximo, acabamos con él de un golpe certero.



¡Puedo volaaaaar!

Para pasar a la siguiente pantalla, nos situamos debajo de donde se encuentra (o se encontraba, según habilidades) el indígena de la plataforma elevada. Para el caso, entre las dos cabañas. Seleccionamos la pertiga, y realizamos un salto con una fuerza de 6.5.

Si la fuerza es la justa, caeremos sobre la primera plataforma de la siguiente pantalla. Sin pararnos un momento, seguimos corriendo hacia la derecha, saltando a la segunda plataforma, y de allí dejándonos caer por la derecha. Esta posición es segura, y podemos emplearla para matar a los indígenas de esta pantalla, aumentando nuestra puntuación.

Cuando nos hayamos cansado, pasamos a la siguiente pantalla, donde tendremos a un mono tirando cocos desde la palmera (granada al canto), y a un cocodrilo al que no podemos matar.



Aunque la entrada a la mina está a la derecha, nos resultará poco menos que imposible burlar al cocodrilo

Si somos pacientes, acabará apareciendo el aguilucho al que mencionamos hace un rato. Esta vez sí, dejamos que nos atrape, llevándonos a su nido, donde recogeremos la segunda joya.

Para escapar del nido solo tenemos que saltar por encima del laguito, salir a la pantalla de la izquierda, esperar a que el tronco pase por debajo, y dejarnos caer sobre él. Conviene tener seleccionado el boomerang, y cargado al máximo, pues al aparecer en la siguiente pantalla, tendremos a un mono en una palmera en el extremo opuesto. Y en este caso, nada mejor que el boomerang para acabar con él.

Una vez eliminado el mono, nos situamos en el extremo izquierdo del tronco, seleccionamos la pértiga, cargamos la fuerza al máximo y saltamos nada más terminar la cascada. Pasaremos a la siguiente pantalla, que resultará ser donde comenzamos la partida, cayendo en esta ocasión sobre la loma central.

Como al principio, eliminamos a la serpiente, pasamos a la loma de la derecha, avanzamos a la siguiente pantalla, usamos la espada para eliminar a la nueva serpiente, hacemos nuestro camino hasta llegar a la loma de la derecha, accedemos a la siguiente pantalla, nos situamos al borde de la loma, matamos a la serpiente de abajo, y nos dejamos caer una vez eliminada esta.

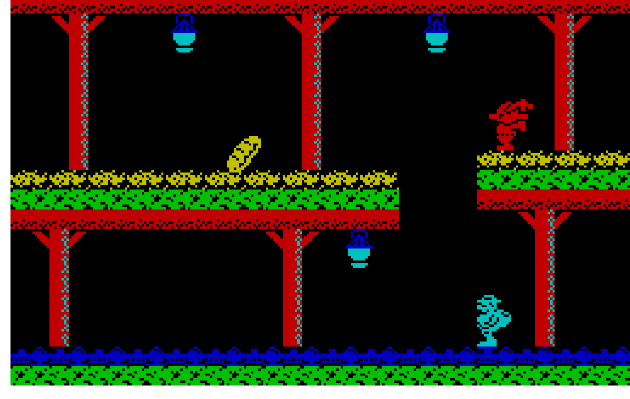
En esta ocasión, en lugar de saltar, regresaremos a la pantalla anterior, esta vez a nivel de suelo, para recoger los víveres. Hay que saltar con cuidado las arenas movedizas, y protegerse de un soldado que aparecerá justo por donde acabamos de entrar.

Regresamos a la pantalla donde nos dejamos caer, con cuidado, pues la serpiente estará allí esperándonos de nuevo. Repetimos el proceso inicial. Es decir, nos situamos a la distancia de una pértiga de la planta carnívora, la usamos (la pértiga) con fuerda 6, eliminamos a los monos de la siguiente pantalla, pasamos a la siguiente pantalla esquivando al aguilucho, matamos al mono de la palmera y al pez del laguito, esquivando siempre al aguilucho (no me cansaré de repetirlo), y subimos al tronco.

¡Cuidado ahora! Esta vez no hay que saltar con la pértiga hacia la derecha. Veremos, bajo la cascada, una especie de gruta. Por ahí es por donde debemos pasar. Si queremos, matamos al mono con el boomerang nada más entrar en la pantalla. Pero lo importante esta vez es situarse bien en el tronco, aproximadamente en el centro, y cuando estemos un cuadro por debajo de la gruta, correr hacia ella y saltar. Si lo hacemos bien, entramos en la caverna que nos conducirá a la mina a la que no pudimos entrar por el cocodrilo.

Una vez en la gruta, lo primero es hacerse cargo del murciélago, haciendo uso de una granada. Con otra granada podemos quitarnos de enmedio al escorpión que hay en el suelo. Y ya con un boomerang a fuerza 5.5, acabamos con el lagarto de la loma superior. Entonces nos dejamos caer a la izquierda, y con la pértiga saltamos sobre la loma, usando entre 5.5 y 6 puntos de fuerza.

Pasamos a la siguiente pantalla, esta vez avanzando hacia la izquierda, y desde el borde de la loma nos despachamos al lagarto y al pez del nivel inferior. Ya sabéis como, con una granadita al canto. Nos dejamos caer, y nos separamos lo justo para que entre la pértiga. Saltamos con el máximo de fuerza, y caeremos sobre un risco en el que se encuentra la tercera joya.



Aunque aquí no se ven, por la mina circulan vagonetas que pueden atropellarnos

Nos dejamos caer desde el risco (está bien eso de caer desde cualquier altura y no hacerse daño), y

pasamos corriendo a la siguiente pantalla, saltando para pasar la estalagmita. Con el boomerang nos deshacemos del escorpión, y seguidamente con la pértiga saltamos a la loma, usando 6 puntos de fuerza.

En la siguiente pantalla, primero damos cuenta del escorpión de la parte superior con un boomerang. Nos dejamos caer sobre el risco, y con una granada hacemos lo propio con el lagarto. Caemos ya a ras del suelo, y con el boomerang acabamos con el escorpión que quedaba. Saltando sin la pértiga pasamos por encima de la estalagmita, recogemos el agua, y regresamos a la posición del risco. Desde allí, usando la pértiga con fuerda de 6, saltamos a la loma de la izquierda.

Antes de pasar a la siguiente pantalla, seleccionaremos el boomerang, y lo cargaremos al máximos. Avanzamos entonces, y nada más entrar, lanzamos el boomerang, acabando con uno de los dos murciélagos. Rápido cambiamos a la granada, y con un poco de fuerza (unos 4 puntos), acabamos con el otro murciélago, y con suerte, con el escorpión. Si este último no ha caído, lo matamos desde el borde con una granada.

Caemos al suelo, y con el boomerang matamos al lagarto. Saltamos la estalagmita, y con la pértiga, a fuerza 4.5-5, saltamos sobre la loma. Seleccionamos el boomerang, lo cargamos al máximo, y entramos en la siguiente pantalla. Matamos al murciélago como hicimos hace un momento, y examinamos la pantalla.



Las sirenas del lago nos aturdirán con sus encantos

Esta pantalla es el acceso a la mina por el lado derecho. Hay una estalactita que señala el punto donde situarnos para, usando la pértiga al máximo, recoger el agua que hay en el saliente de la viga. Conviene esperar, pues sale un soldado, que si tenemos mucha prisa, puede coincidir su disparo con nuestra caída. Así que mejor, desde el suelo, agachados, esperamos al soldado, lo matamos, y saltamos a por la jarra.

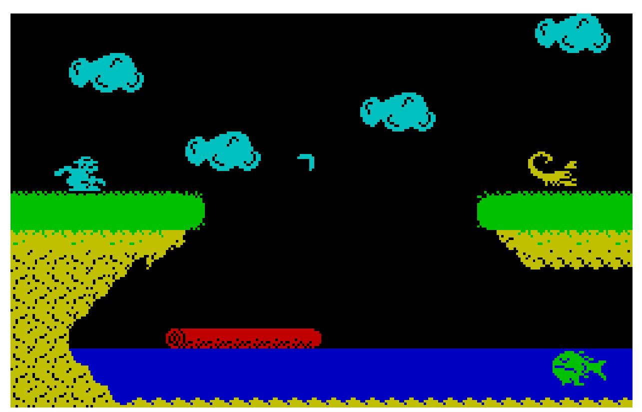
Antes de continuar, una advertencia. Por la mina circulan carros sueltos, que pueden aparecer por cualquier lado. Podemos saltar por encima de ellos sin demasiada dificultad, aunque requiere algo de práctica. Si estáis pensando en grabar un snapshot de la partida para retomar luego posiciones por si os matán, este es el mejor momento.

Pasamos las pantallas hacia la izquierda, observando que hay dos niveles. No perdamos el tiempo con los mineros que hay en el nivel superior. Sigamos caminando hasta llegar a una pantalla donde el nivel superior tiene un hueco en el centro por el que podemos pasar saltando con la pértiga.

Primero iremos hacia la izquierda. Así que nos situamos bajo el minero de la derecha. Antes de saltar, procuraremos eliminar al menos al minero de la izquierda. Siempre cuidándonos de las vagonetas, con el boomerang aproximaremos el tiro (de unos 5 puntos de fuerza). Cuando le matemos, saltamos con la pértiga usando 8 puntos de fuerza. ¡Ojo! También hay que esquivar las nubes de gas.

Una vez en el nivel superior, pasamos a la pantalla de la izquierda y nos agachamos nada más entrar. Estos mineros son más agresivos que los indígenas, y nada más vernos empiezan a lanzarnos sus herramientas. Agachados como estamos, cargamos el boomerang al máximo, y cuando haya un hueco entre disparos enemigos, lo disparamos. Con el camino despejado, recogemos la que ya es nuestra cuarta joya. Regresamos a la pantalla anterior, cuidando de eliminar a los mineros con el procedimiento indicado para esta pantalla.

Nos dejamos caer, y nos situamos justo en el poste de la izquierda. De nuevo con la pértiga a unos 8 puntos, saltamos, y llegamos al nivel superior del lado derecho. Hacemos ahora nuestro camino hacia la derecha, eliminando a cuanto se nos cruza en el camino, y saltando el mini hueco que hay en una de las pantallas.



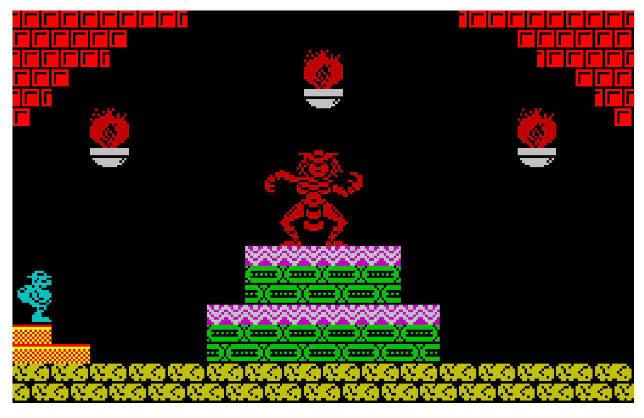
Acción complicada: Caer en el tronco y saltar con la pértiga a donde se encuentra el escorpión. De lo contrario, el tronco nos llevará al principio del juego

Llegamos a una zona que he dado en llamar el lago, por su gran extensión de agua. Nos agachamos y con el boomerang matamos al hombre que nos ataca. En el agua aparecen unas sirenas, que nos lanzan corazones. Estos ataques no nos restan vida, pero nos aturden lo suficiente como para sufrir problemas (el aguilucho, caídas al agua, etc.). Cuando el tronco este a nuestro alcance, saltamos sobre él, y con la pértiga pasamos al camino superior de la derecha.

Como antes, al acceder a esta nueva pantalla tenemos a un hombre al otro extremo que nos ataca, y que tenemos que acabar con él. Otra vez recurrimos al tronco, en esta ocasión prestando especial cuidado de no olvidarnos de la quinta y última joya. Saltamos al siguiente camino, y seguimos hacia la derecha, sorteando enemigos y agujeros en el camino. Llegamos a un punto donde se ve a la derecha tierra firme. Como hasta ahora, nos deshacemos del hombre, y usamos el tronco para llegar a la orilla. Saltamos con la pértiga, pero con cuidado de no pasarse. La fuerza debe ser no superior a 5 puntos.

En la siguiente pantalla vemos que aparece de nuevo el agua, con un tronco, y algunos animales. Acabamos primero con el escorpión (arma a vuestra elección), y con el pez (granada, por supuesto). Esperamos a que aparezca un nuevo tronco, con la pértiga seleccionada, y caemos sobre él. Nada más caer nos situamos en el extremo izquierdo, cargamos fuerza (unos 5.5 puntos) y saltamos. Hay que hacerlo rápido, pues de lo contrario el tronco nos llevará al inicio de la partida.

Pasamos a la siguiente pantalla, dejándonos caer al suelo, y, agachados, matando a los dos indígenas. Nos situamos a una distancia de una pértiga de la piedra, y con una fuerza de 8 puntos, saltamos por encima de la planta carnívora, pasando de largo la pantalla que hay a continuación, y llegando a la entrada del templo, donde nos esperan dos nuevos indígenas. Agachados, rápidos y cautelosos, acabamos con ellos con el boomerang, y entramos al templo antes de que el aguilucho haga de las suyas.



El guardián del templo nos dejará pasar si le llevamos las cinco joyas que se encuentran desperdigadas por la selva

Veremos entonces al guardián de la entrada hacer una especie de baile, en el que le entregamos las joyas que hemos recogido. Tienen que ser cinco, si son menos, nos reenvia a donde nos falten recogerlas.

Una vez en el templo, tendremos que abrirnos paso entre trampas, indígenas y pruebas de habilidad varias.

Ya en la primera pantalla observamos que el suelo se mueve siempre en dirección contraria a la que nosostros caminamos. En el suelo hay unas marcas de color azul. Cuando las pisamos, se lanzan unas flechas desde unas trampas situadas en el techo. Pero antes de tratar de esquivarlas, tenemos que retirar los dos muros que nos bloquean el paso. Como ya es habitual, hay que demostrar nuestra habilidad con el boomerang.

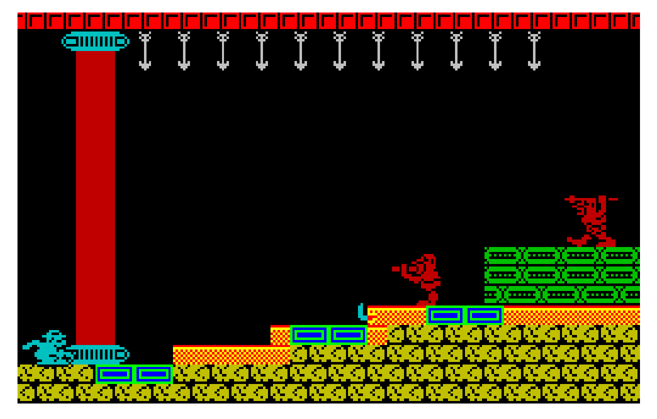
Sitados justo al principio de la pantalla, lanzamos el boomerang hacia la derecha con la intención de accionar las palancas del techo. A la primera palanca la alcanzamos con una fuerza de 4.5, mientras que para la segunda palanca necesitamos 5 puntos de fuerza exactos.

Una vez retirados los muros, pasamos a la siguiente pantalla, esquivando las marcas azules del suelo.

En esta nueva pantalla, nos quedamos de nuevo nada más entrar, agachados. Con el boomerang a fuerza 5 eliminamos a uno o dos de los indígenas (depende de lo próximos que quedemos a los cinco puntos de fuerza).

Si no hemos eliminado al segundo indígena, el de arriba del todo, no pasa nada. Nos acercamos al borde de la losa azul, sin pisarla, y con la granada podemos alcanzarle. Para protegernos de sus lanzas, basta con agacharnos.

Cuando tengamos el camino despejado, seleccionamos la pértiga y saltamos con una fuerza de 8.8 puntos, pasando de largo la siguiente pantalla y aterrizando sobre una losa trampa de la pantalla que hay más allá. No pasa nada, no hay flechas a esa altura. Recuperamos una posición segura, y nos agachamos para protegernos de la lanza del indígena. De nuevo con el boomerang, a fuerza 5.5, nos quitamos de enmedio a los dos indígenas.



No hay nada como el boomerang para acabar con los indígenas

Para subir a la plataforma verde, usamos la pértiga con fuerza 6, lo que nos mantiene en la misma pantalla.

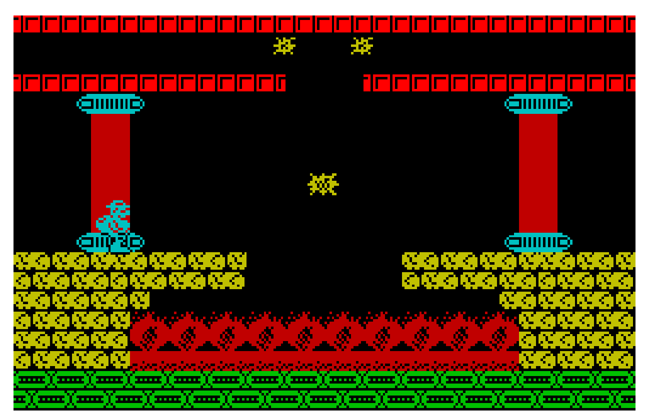
La siguiente pantalla tiene una cierta complicación, porque tendremos a dos indígenas atacándonos con sus flechas, y a un brujo que se mueve por la pantalla, y que no se queda corto en sus ataques. La técnica que mejor me ha funcionado para pasar es entrar corriendo, saltar, lanzando en el aire un boomerang con la máxima fuerza, y seguir corriendo hasta la siguiente

pantalla.

En la siguiente pantalla tenemos tres plataformas, separadas por sendos huecos, y una barra de pan muy tentadora en el suelo, con el cocodrilo. Si tenemos suficientes vidas, podemos pasar de la barra de pan, que luego resulta difícil salir del hoyo.

Pasamos a la siguiente pantalla, saltando de plataforma en plataforma, y llegamos a una pantalla con un hueco en el centro, fuego en la parte inferior, y un agujero en el techo por el que caen bolas de fuego. Para colmo, aparecerá en algún momento un soldado en la parte derecha. Esperamos al soldado, lo matamos, y desde la primera columna, hacemos nuestro cálculo para determinar cuando es el mejor momento de saltar entre las bolas de fuego que caen. No hay regla fija, aunque lo mejor es empezar a correr cuando veamos que la última bola de fuego de está a punto de caer, saltar el hueco, y continuar hacia la derecha.

Este es otro punto clave del juego. De nuevo, como ya os comenté antes, si tenéis intención de asegurar posición con un snapshot, este es el mejor momento.



Hay que calcular el salto para pasar entre las bolas de fuego

La siguiente pantalla es la salida del templo. De nuevo en espacio abierto, estamos a merced de la aparición de un aguilucho que nos llevaría de regreso al nido, y por tanto, a repetir toda la partida desde el principio. Creo que estamos todos de acuerdo en que esto no es algo deseable.

Bueno, nos deshacemos de la serpiente que hay en el suelo con una buena granada, con fuerza entre 5 y 6 puntos. Nos situamos al borde de la escalera, y saltamos con la pértiga con fuerza 6. Con ese santo lograremos caer justo en el agujero de la siguiente pantalla, entre las dos plantas carnívoras, evitando al mono, y lo que es más importante, quitándonos de la preocupación de que aparezca el aguilucho.

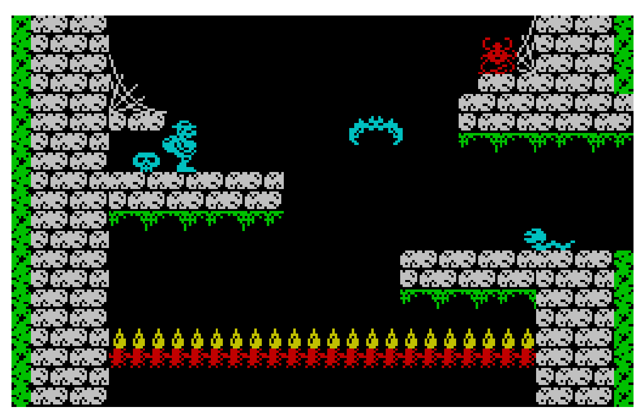
Caemos por una gruta, en la que nos van a asediar los murciélagos. Granada en mano, nos deshacemos de la araña del risco de la izquierda (fuerza 5.5), y si el murciélago no ha subido mucho, nos olvidamos de él, saltando a donde estaba la araña.



Tenemos que caer por el pozo para entrar en la gruta

Pasamos a la pantalla inferior (nos dejamos caer, vamos), y con un boomerang o con una granada, matamos al murciélago que nos viene de frente. Ahora sí, con la granada, acabamos con la serpiente, y saltamos donde estaba. En la pantalla de la derecha tenemos una palanca que se lleva la empalizada de lanzas que en este pantalla. Así que entramos a la derecha, y boomerang en mano, accionamos la palanca (fuerza 3.2). Tenemos que esperar a que aparezca la empalizada en esta pantalla, y entonces regresamos a la anterior, cuidándonos de la serpiente (boomerang o espada), y del murciélago (granada).

Saltamos desde el borde hacia la izquierda, cayendo sobre un risco en la pantalla inferior. Damos cuenta del murciélago y de la araña de la derecha. De nuevo saltamos desde el borde, esta vez hacia la derecha, cayendo donde estaba la araña. Caminamos hacia la izquierda, cayendo en la pantalla inferior.



Para quitar la empalizada tenemos que accionar la palanca que hay en la estancia contigua

En esta pantalla tenemos de nuevo una empalizada por debajo, con lo que intuimos que en la estancia de la izquierda debe estar la palanca para retirarla. Primero matamos a la araña y al murciélago. Echamos a correr hacia la izquierda, saltando desde el borde donde estaba la araña. Pasamos a la siguiente estancia, y accionamos la palanca con el boomerang (fuerza 4.5).

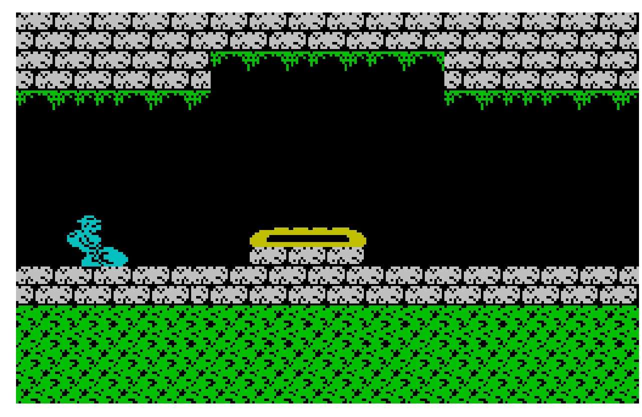
Nos dejamos caer por el hueco. Matamos al escorpión y al murciélago. Toca saltar hacia la

derecha, por encima de la laguna, y seguir caminando hacia la derecha, saltando por encima de las piedras.

Seguimos hacia la derecha hasta que nos encontramos con un pozo. Este es otro punto negro, en el que además, si nos caemos por el pozo, podemos dar por perdida la partida, pues no hya forma de salir de él. La forma de sortearlo es usar la pértiga, próximos al pozo, con una fuerda de entre 3.5 y 4 puntos.

Pasado el escollo del pozo, llegamos a una escalinata con dos murciélagos. Los matamos, y nos situamos en el peldaño en cuyo borde hay una piedra. Gracias a la piedra tenemos longitud suficiente para usar la pértiga. Saltamos con la pértiga con 8 puntos de fuerza, cayendo en la entrada de un nuevo templo.

Hay que eliminar a los indígenas, teniendo mucho cuidado de que el aguilucho no nos capture.



¡Mucho cuidado con este pozo! Si caemos en él no podremos salir

Entramos en el templo y allí encontraremos cómodamente sentado a quien estabamos buscando... ¿o tal vez no? ;-P ✍

Livingstone, supongo...

LINKS

- Descarga la solución completa en formato RZX: http://www.speccy.org/magazinezx/revistas/9/rzx/livingstone_rzx.zip
- Descarga el mapa de la aventura: http://www.speccy.org/magazinezx/revistas/9/img/livingstone_mapa.zip



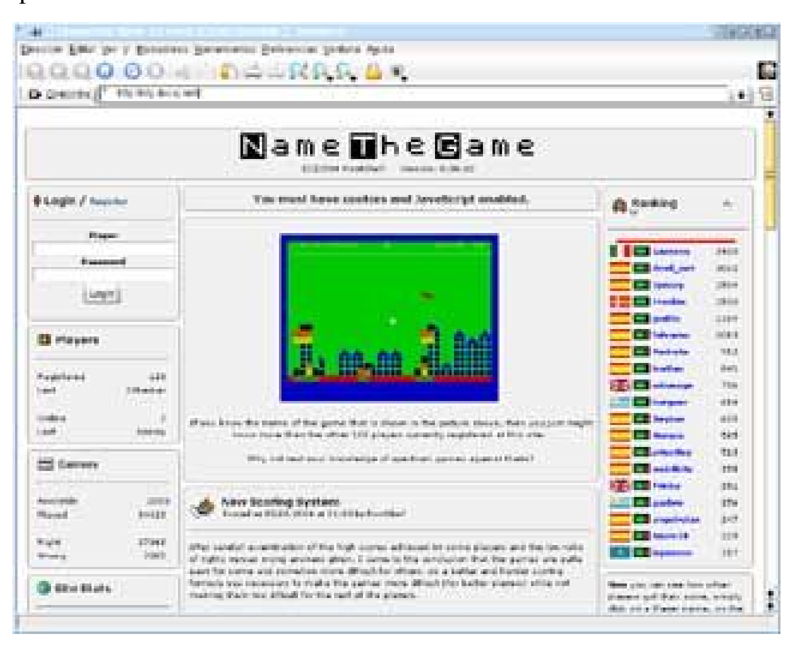
SIEW

NAME THE GAME

Si en el mundo del Spectrum, sobre todo en la escena española, existen las modas, es algo que merecería un estudio un día de estos.

Está claro que el conjunto de aficionados al Spectrum está formado por personas más o menos normales y que, por lo tanto, pueden verse atraídas por modas pasajeras, como el resto de los mortales, pero... ¿modas relacionadas con el Spectrum? La web que analizamos en esta ocasión, 'Name the game', está de moda. Sólo con visitarla podremos observar como es seguida por un gran número de entusiastas españoles, conocidos en es.comp.sistemas.sinclair. ¿Qué tiene esta web para ser tan atrayente?

Lo que está claro es que 'Name the game' engancha. El reto que propone es tan simple como adivinar el título del juego al que pertenecen las capturas de pantalla que se nos van mostrando, entre una serie de opciones. Es algo tan simple que sorprende que no se le haya ocurrido a nadie antes.

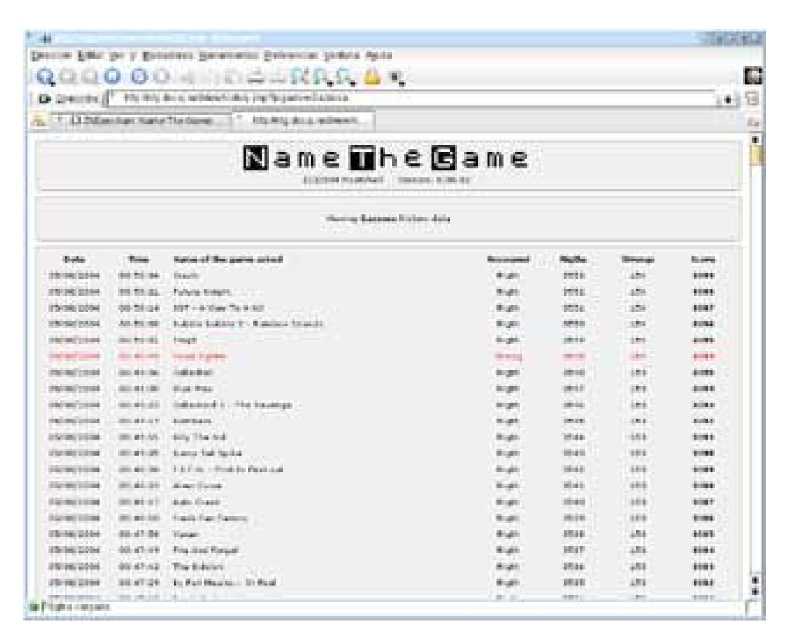


Punto de entrada a la web

DISEÑO

Al abrir la página con nuestro navegador, veremos también la simpleza de su diseño. Si no estamos registrados todavía (ya hablaremos de ello más tarde), podremos distinguir tres zonas principales, utilizando un color 'blanco oscuro' de fondo y un color negro para la fuente, a modo de pantalla del televisor cuando se usa un Spectrum. En la parte central, podremos observar una pantalla de ejemplo (que es diferente cada vez que accedemos a la web), con un mensaje retándonos a que nos enfrentemos a las personas ya registradas hasta el momento (a la hora de escribir esto, son un total de 128 los registros contabilizados). Siguiendo en la parte central, y debajo de esta pantalla de ejemplo, vemos las noticias relacionadas con el sitio. Todo el diseño nos recuerda a cualquier página creada con un gestor de contenidos como Drupal o Geeklog (un ejemplo de web generada con este tipo de aplicaciones es speccy.org).

En la parte derecha tenemos el ranking. Es increíble ver como la inmensa mayoría de los jugadores en primeras posiciones son españoles (y muy conocidos para los habituales de las news de Spectrum en español). Podremos ver la nacionalidad de cada uno, una barra de color verde y rojo que nos indica la proporción de aciertos con respecto a la de fallos, el nombre del jugador y su puntuación. Si además pulsamos sobre el nombre de un jugador podremos ver el historial de aciertos y fallos de dicho jugador (la fecha y la hora en la que se participó, el nombre del juego preguntado, si fue un acierto o un fallo, el número de aciertos y fallos hasta el momento, y la puntuación, remarcando en color rojo las líneas correspondientes a los fallos). Esta página de estadísticas, según el autor del sitio web, propicia el que se pueda comprobar si determinado jugador hace trampa haciendo uso de algún programa que se conecte automáticamente a la red y conteste muy rápido. Es algo que se ha añadido de forma reciente y que todavía no sabemos si permitirá



Ejemplo de estadísticas de un jugador determinado

cumplir el objetivo de detección de trampas.

En la parte izquierda de la pantalla principal tenemos varios cuadros con información. En el primero podremos acceder al juego si ya somos usuarios registrados, o registrarnos en caso contrario. En el segundo podemos ver



¿Dónde tengo que pulsar para registrarme?

el número de jugadores registrados hasta el momento, así como el alias del último que lo hizo, cuántos hay conectados en ese instante, y el alias del último usuario registrado que se conectó a la página. El tercer cuadro indica la cantidad de juegos disponibles en la base de datos (de momento 1000), cuántas veces se ha jugado, cuántas veces se ha acertado y cuántas se ha fallado. En el cuarto podremos ver algunas estadísticas del sitio.

El siguiente cuadro es altamente interesante. Cada vez que entremos en la página se nos mostrará un enlace aleatorio a alguna página sobre el Spectrum (no sólo páginas inglesas; en una de las ocasiones el autor de este comentario pudo ver en este cuadro un enlace a la página de Horace comentada en el número anterior de MagazineZX). Por último, en el cuadro situado al final de esa columna de la parte izquierda podremos acceder al típico libro de visitas.

REGISTRÁNDONOS

Registrarse es muy sencillo... ¡y además gratis!. Debemos comenzar pulsando sobre el enlace 'register' situado en el primero de los cuadros situados en la parte izquierda de la página principal, tal como muestra la siguiente imagen:

La pantalla de registro es muy curiosa. De hecho, aunque a lo mejor no es tan raro, el autor nunca había visto nada similar. Además del formulario de registro, podremos consultar a la izquierda un listado con TODOS los jugadores que se hayan apuntado hasta el momento. Examinando este listado vemos como otros habituales de es.comp.sistemas.sinclair que no aparecen en la lista de jugadores con más puntuación también están apuntados. Los nombres aparecen en tres colores, azul si es masculino, rosa/violeta si es femenino, y negro (es de suponer que son jugadores de los que no se dispone información sobre su sexo porque se este dato se emepezó a pedir más tarde de que ellos se registraran).

En el formulario deberemos introducir un alias y una contraseña para acceder a la página, una dirección de correo, nuetro sexo y nuestro país de procedencia. Tras completar el registro, volveremos a la página principal, desde donde podremos acceder introduciendo nuestro alias y contraseña en la parte izquierda.

JUGANDO

Una vez entramos con nuestro alias y contraseña, son dos los cambios significativos que vemos en la página, con respecto a la página principal. Por una parte, en el primer cuadro de la parte izquierda, los campos del formulario para introducir alias y contraseña desaparecen para mostrarnos información sobre nuestra propia cuenta, como nuestra puntuación, el número de juegos que hemos intentado adivinar, nuestros aciertos, y nuestros fallos. También dispondremos de dos enlaces, 'Logout' para deshabilitar nuestro acceso, y 'Account'. para cambiar los datos de nuestra cuenta. Sobre estos dos enlaces una barra, con colores verde y rojo nos mostrará la proporción de aciertos y fallos que hemos tenido. Con respecto a la columna central, todas las noticias desparecen, y aperece la pantalla del juego que debemos adivinar, con cinco posibles respuestas, de las que debemos



El formulario de registro y la lista de jugadores ya apuntados

seleccionar una. Como se puede leer en la página principal, conforme nuestra puntuación es mayor, el número de respuestas crece, para hacer el reto todavía más atrayente. Es curioso cómo justo debajo de la captura de pantalla del juego del cual tenemos que adivinar su nombre aparece otra barra de colores verde y rojo que nos muestra la proporción de veces que el resto de jugadores han acertado o fallado a la hora de adivinar a qué juego pertenece.

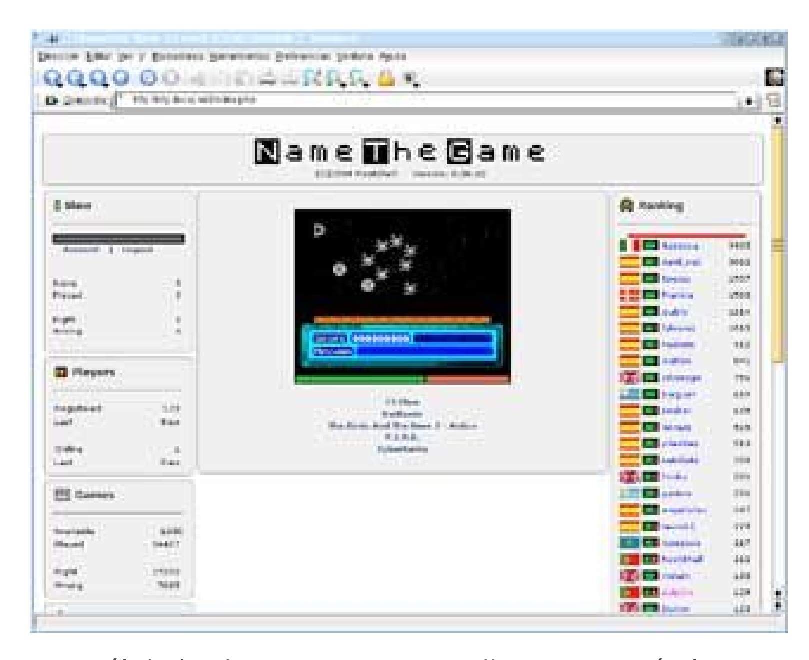
Al dar alguna respuesta veremos otra de las medidas antitrampa: una espera de un segundo entre respuesta y respuesta. Esto que en un principio puede parecer una buena solución, tampoco perjudica demasiado a los tramposos, en opinión del autor de este texto, pues todavía se podría dejar un programa jugando de forma automática, solo que en lugar de respuestas inmediatas daría una respuesta cada segundo. Otra medida antitrampas consiste en que cuando el jugador pulsa sobre logout y se vuelve a identificar más tarde poniendo su alias y contraseña, el juego que se le preguntaba antes del logout será el mismo que se le preguntará tras volver a acceder, de tal forma que se evita que ante un juego que no conoce, el jugador intente salir y volver a entrar para que le pregunten otro.

CONCLUSIONES

Muy divertido. Mientras elaboraba este comentario yo mismo me he visto enganchado y he tenido que parar en varias ocasiones para jugar un rato. Con respecto a la web en sí misma, el diseño es austero pero agradable, con la información justa. Más florituras la sobrecargarían y entonces se perdería parte de la gracia.

Se nos ocurren varias sugerencias. Por ejemplo, que en el ranking de la página principal aparezca el número de puesto en el que se encuentra el jugador. Esto para las primeras posiciones no es tan importante, pero cuando empiezas a consultar la parte media de la tabla o la cola, te pierdes y al final no sabes en qué posición está cada jugador. ¿Todavía no lo has probado? Entonces no estás en la onda...✍

PUNTUACIÓN: 7



¿De cuál de los juegos que se nos dice se tratará el que nos muestra el pantallazo?

LINKS

- Name The Game : http://ntg.docaj.net/
- speccy.org : http://www.speccy.org/



SIEW

CREANDO UNA AVENTURA CONVERSACIONAL CON Z88DK (III)

Terminamos en esta entrega con la explicación de las herramientas de texto de la librería z88dk aplicadas a un juego en concreto, nuestra aventura conversacional del pirata Guybrush Threpwood. Desde luego, con todo lo explicado no se pretende que el resultado final sea una aventura conversacional completa, eso conllevaría un trabajo adicional por parte del lector de la serie, sino que se sienten las bases para aprender a manejar estas herramientas de texto mediante un ejemplo práctico.

A partir de la siguiente entrega dejaremos atrás la parte de la librería referente al texto y pasaremos a los gráficos.

¿QUÉ NECESITAMOS?

Lo mismo que se ha venido utilizando hasta el momento: el archivo aventura.c con el código fuente de nuestro juego, el archivo datos.h con las estructuras de datos que vamos a utilizar y el archivo Makefile, en el caso de que lo estemos utilizando.

CREANDO UN PIRATA QUE SE MUEVE POR LA TABERNA

Una parte muy importante de las aventuras suelen ser los PSI o *Personajes Pseudo Inteligentes* (tal como son llamados en el PAWS o en las aventuras de AD, por ejemplo). Básicamente, un PSI se limitará a deambular por alguna parte del mapeado de la aventura (aunque también podría estar fijo en una localización) y a responder a ciertas acciones que hagamos, e incluso podría hablar con nosotros.

Lo que pretendemos en esta entrega es la creación de un pirata que se mueva por la taberna (y sólo por la taberna, no saldrá fuera de ella ni irá al callejón), con el que podremos hablar, darle la jarra para que se emborrache, e incluso pelear con la espada.

La única información que tenemos que almacenar para que nuestro pirata cumpla con nuestros objetivos es la habitación en la que se encuentra en ese momento y si ha bebido de la jarra (y, por lo tanto, está borracho). Esto lo haremos mediante el uso de dos variables declaradas al inicio del método main (las líneas en rojo, como en anteriores ocasiones, nos indican cambios respecto al código de anteriores entregas):

```
void main(void)
{
   int habitacion = 0;
   int final = 0;
   char comando[250];
   int i;
   int hayObjetos;
   char palabra[50];
   int pesoTransportado;
   int localizacionPirata = 5;
```

La variable localizacionPirata nos va a indicar en qué habitación se encuentra el pirata en cada momento. Hemos hecho que su localización inicial sea la que se encuentra en la posición 5 del array de habitaciones (el reservado de la taberna). Este valor podrá cambiar aleatoriamente a lo largo de la aventura. Tras cada comando del jugador, se determinará al azar si el pirata cambia o no de habitación y hacia cuál. La variable borrachoPirata indica si el pirata está borracho porque le hemos dado la jarra. Solo le asignaremos el valor 0 si no lo está o el valor 1 si lo está; es por ello que empleamos un tipo short int, pues no vamos a necesitar reservar más memoria.

Lo siguiente que vamos a escribir es la parte del código que determina aleatoriamente si el pirata se va a mover o no y hacia donde. Todo esto lo introducimos dentro de una función de tal forma que luego podremos ponerlo en la parte que queramos. La función, que debe ser definida antes de main, tendría la siguiente forma:

```
int movimientoPirata(int localizacionPirata, THabitacion
habitaciones[],
  int habitacion)
{
  int probabilidadMovimiento;
  int localizacionAnterior = localizacionPirata;

  probabilidadMovimiento = rand()/(RAND_MAX/4);

  if (habitaciones[localizacionPirata].direcciones
[probabilidadMovimiento] > 2)
  {
    localizacionPirata =
      habitaciones[localizacionPirata].direcciones
[probabilidadMovimiento]-1;
```

```
    if (localizacionAnterior != localizacionPirata)
    if (localizacionPirata == habitacion)
      printf("El pirata entra en la habitacion.\n\n");
    else if (localizacionAnterior == habitacion)
    {
      printf("El pirata sale de la habitacion hacia el ");
      if (probabilidadMovimiento == 0) printf("norte.");
      else if (probabilidadMovimiento == 1) printf("este.");
      else if (probabilidadMovimiento == 2) printf("sur.");
      else printf("oeste");
      printf("\n\n");
    }
  }

  return localizacionPirata;
}
```

En un momento explicaremos qué hace todo este código, pero antes, un apunte. Ya que vamos a utilizar la función *rand()*, hemos de tener en cuenta que esta función se encuentra definida en el interior del archivo de cabecera stdlib.h, que no hemos incluido hasta el momento. Por ello, deberemos añadir la siguiente línea al inicio del archivo, junto al resto de las cláusulas #include:

```
   #include <stdlib.h>
```

Una vez aclarado esto, continuamos. La función recibe como parámetro la localización actual del pirata, el array de habitaciones, y la habitación actual en la que se encuentra el jugador. Recordemos que tanto el primer como el tercer parámetro indican posición dentro del array de habitaciones. La función devolverá un entero que será la nueva localización del pirata (tanto si se ha movido como si no).

Lo primero que se hace en la función es declarar dos variables, probabilidadMovimiento, que almacenará en qué dirección intenta moverse el pirata, y localizacionAnterior, que contiene la localizacion del pirata tal como se encuenta al iniciar la función, para guardar el valor anterior en el caso de que éste cambie.

La siguiente línea es la que determina al azar en qué dirección intentará moverse el pirata. Hemos de tener en cuenta que hemos decidido que el pirata sólo se podría mover por el interior de la taberna y, por supuesto, sólo podrá moverse en direcciones correctas, pero esto lo veremos más adelante. Para obtener números aleatorios, se hace uso de la función rand(). Esta función se llama sin parámetros y devuelve un entero aleatorio entre 0 y RAND_MAX, que tiene un valor de 32767 (según está definido en stdlib.h). Sin embargo, a nosotros nos interesa obtener un valor entre 0 y 3 (los posibles índices del array de direcciones dentro de la estructura THabitacion). Es por ello que recurrimos al truco que se muestra en el código.

Si en Linux ejecutamos *man rand*, veremos que allí se nos explica una técnica para poder calcular números aleatorios en un rango determinado (de 1 a 100, por ejemplo). Sin embargo, no podemos hacer uso de esta forma con z88dk, pues no se admiten los números de coma flotante. Tampoco podríamos multiplicar rand() por cuatro y después dividir, porque podríamos desbordar la capacidad del entero; es por ello que calculamos el entero entre 0 y 3 de la forma tan rebuscada que se muestra en el código. Si quisieramos que fuera de 1 a 4 sería igual, pero sumando 1 al total. Deberemos recordar esto siempre que queramos calcular un número aleatorio en un rango determinado (por ejemplo, una tirada de dado).

En la siguiente línea, correspondiente al if, es donde controlamos que el pirata se mueva si y solo si el momvimiento es correcto.

```
if (habitaciones[localizacionPirata].direcciones
[probabilidadMovimiento] > 2)
```

Dentro de la habitación correspondiente en el array de habitaciones cuyo índice se corresponde con la localización del pirata, comprobamos el valor de habitación a la que se movería el pirata según la dirección calculada al azar y almacenada en probabilidadMovimiento. Sólo realizaremos las siguientes acciones (que se explicarán a continuación) si este valor es mayor de 2. ¿Por qué esto es así? Porque si valiera 0, sería un movimento incorrecto (no hay habitación en esa dirección), y los valores 1 y 2 se corresponden con las habitaciones del juego situadas fuera de la taberna, por donde habíamos acordado anteriormente que el pirata no se podría mover.

Si esta condición se cumple, se ejecutará el siguiente trozo de código:

```
  localizacionPirata =
    habitaciones[localizacionPirata].direcciones
[probabilidadMovimiento]-1;
  if (localizacionAnterior != localizacionPirata)
    if (localizacionPirata == habitacion)
      printf("El pirata entra en la habitacion.\n\n");
    else if (localizacionAnterior == habitacion)
    {
      printf("El pirata sale de la habitacion hacia el ");
      if (probabilidadMovimiento == 0) printf("norte.");
      else if (probabilidadMovimiento == 1) printf("este.");
      else if (probabilidadMovimiento == 2) printf("sur.");
      else printf("oeste");
      printf("\n\n");
    }
```

Primero hacemos que la nueva localización del pirata sea a la que se movería según la dirección calculada al azar (restamos uno, porque el array direcciones almacena identificadores de habitación, no índices dentro del array de habitaciones). Lo siguiente es informar al jugador de que el pirata ha entrado o ha salido de la habitación en la que actualmente se encuentra. Como el pirata se ha movido, si la habitación actual del pirata es la misma que la del jugador, eso quiere decir que el pirata ha entrado en la misma habitación en la que estaba el jugador, por lo que mostramos el mensaje adecuado. Por otra parte, como el pirata se ha movido, si la localización anterior del pirata era igual a la habitación en la que se encuentra el jugador, eso quiere decir que el pirata ha salido de esa habitación, por lo que mostramos un mensaje indicando hacia donde ha salido, según el valor aleatorio de probabilidadMovimiento.

Lo último que hace la función es devolver el valor

de localizacionPirata, tanto si ha cambiado como si no.

Si ahora probáramos nuestra aventura, todavía no pasaría nada, porque no hemos incluido esta función en ningún lugar del código de main. Parece claro que si queremos determinar si el pirata se mueve después de cada movimiento del jugador, deberíamos llamar a esta función dentro del intérprete de órdenes, pero... ¿dónde?. Un buen lugar sería justo al comienzo de dicho intérprete de órdenes (el código en rojo es código nuevo):

```
escribirDescripcion(habitaciones,habitacion,objetos);
while (final == 0)
{
  localizacionPirata =
     movimientoPirata
(localizacionPirata,habitaciones,habitacion);
  printf("Que hago ahora? - ");
  gets comando;
```

Fuse
File Options Machine Media Help

El pirata sale de la habitacion haci
a el sur.

Que hago ahora? - s

El salon principal de la taberna. De
cenas de piratas se encuentran aqu
bebiendo, cantando, peleandose, vomi
tando, y metiendo mano a las doncell
as del lugar. Si te diriges al norte
, este u oeste podras visitar varias
 de las habitaciones del local.

Salidas: Norte Este Sur Oeste

En la habitacion puedes ver: nada

El pirata esta aqui.

El pirata sale de la habitacion haci
a el oeste

Que hago ahora? -

El pirata es muy juguetón y no para de moverse...

Evidentemente, el valor devuelto por movimientoPirata deberá ser almacenado en la variable localizacionPirata para que los cambios tengan lugar. Si ahora jugamos nuestra aventura podremos ver como a veces se nos informará de las entradas o salidas del pirata, pero no sabremos nunca a ciencia cierta donde se encuentra... falta un detalle... que se nos indique que el pirata se encuentra en la misma habitación que nosotros en el caso de que esto sea así, aunque no se mueva.

Por lo tanto, al final de la función escribirDescripción, que mostraba por pantalla la descripción de la habitación en la que nos encontrábamos, debemos añadir las líneas en rojo:

```
  if (hayObjetos == 0)
     printf(" nada");
  printf("\n\n");
  if (habitacion == localizacionPirata)
     printf("El pirata esta aqui.\n\n");
```

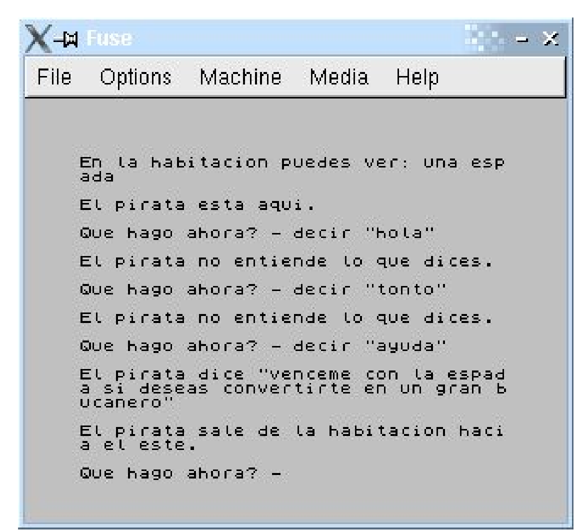
Simplemente comprobamos si la localización actual del pirata se corresponde con la habitación en la que se encuentra actualmente el jugador, y en caso de que sea así, se muestra un mensaje por pantalla. Introducir este cambio implica pasar un nuevo parámetro a la función escribirDescripción:

```
void escribirDescripcion
(THabitacion habitaciones, int habitacion,
TObjeto objetos, int localizacionPirata)
```

Y por supuesto, añadir este nuevo parámetro a todas las llamadas a la función que se hagan dentro de main(). Si jugamos ya tendremos un pirata autista moviéndose por nuestro mundo, incapaz de comunicarse con nosotros y que no reacciona ante ninguna de nuestras acciones. Las siguientes dos secciones tratarán de cambiar esto.

INTERACTUANDO CON EL PIRATA

Dar la capacidad al jugador de hablar con los PSI es algo complicado, pues sería necesario implementar un analizador sintáctico que permitiera al juego interpretar las palabras del jugador. Sería como un intérprete de comandos dentro del intérprete de comandos, pero algo más complicado. Esto, desde luego, queda fuera del objetivo de este artículo. Sin embargo, puede ser interesante indicar el comienzo del camino a seguir para que sea el propio lector el que pueda añadir esta funcionalidad de forma elaborada a su aventura, y a la vez hacer un par de anotaciones sobre características de las cadenas en z88dk.



El pirata parece un poco tonto, pero sabe más de lo que dice...

Vamos a hacer que el jugador le pueda decir cosas al pirata, pero que éste sólo entienda la palabra "ayuda". Si el jugador dice "ayuda" al pirata, éste le dirá qué es lo que tiene que hacer para terminar el juego (vencerle con la espada). Si el jugador dice cualquier otra cosa, el pirata no le entenderá. Como en la mayoría de las aventuras, el jugador deberá teclear algo similar a lo siguiente para poder decirle algo a un PSI:

decir "frase"

Como sólo hay un PSI en nuestra aventura vamos a

obviar el 'decir a', de forma que todo será más sencillo. De todas formas, si nos fijamos, lo que dice el jugador se encuentra entre comillas. Cuando tengamos que analizar el comando introducido por el jugador en el intérprete de comandos, tendremos que hacer uso de cadenas que almacenan comillas en su interior... ¿cómo podemos hacer esto sin que nos dé un error?. Si nosotros tecleamos en un programa *printf (""hola"");* esperando que la salida sea "*hola*", nos llevaremos un chasco porque no funcionará. Deberemos hacer uso de las *secuencias de escape*.

En realidad, en entregas anteriores ya hemos hecho uso de algunas secuencias de escape (por ejemplo, \n ó \0). Una secuencia de escape es un símbolo de barra invertida \ seguido de un carácter. Su misión es introducir un carácter especial dentro de una cadena. Por ejemplo, la secuencia de escape \n introduce un salto de línea en la cadena, la secuencia de escape \a (en C estándar) hace sonar un pitido, etc. Si queremos introducir unas comillas dentro de una cadena para que formen parte de esa cadena (y no se interpreten como delimitador de la misma), deberemos hacer uso de la secuencia de escape \". De esta forma, si quisiéramos escribir "*hola*" por pantalla, la instrucción que debemos incluir en nuestro programa podría ser *printf(*"\"*hola*\""*)*. El primer y último carácter de comillas son los delimitadores de la cadena, mientras que los símbolos de comillas precedidos por \ se escribirán por pantalla.

Dicho esto ya podemos introducir en nuestro intérprete de comandos las sentencias adecuadas para que el jugador pueda dialogar (de forma limitada) con el pirata. Introducimos el siguiente código justo después del comando "apagar antorcha" y justo antes de los comandos de más de una palabra (el código en rojo es el que se introduce):

```
   }
   else if (localizacionPirata == habitacion)
   // comandos del pirata
   {
      strcpy(palabra,strtok(comando," "));
      if (strcmp(comando,"decir") == 0)
      {
         strcpy(palabra,strtok(0,"\0"));
         if (palabra == 0)
            printf("\n\nQue es lo que quieres decir?\n\n");
         else
         {
            if (!strcmp(palabra,"\"ayuda\""))
               printf("\n\nEl pirata dice \"venceme con la
espada si deseas convertirte en un gran bucanero\"\n\n");
            else
                   printf("\n\nEl pirata no entiende lo que
dices.\n\n");
         }
      }
   }
   else
   // Comandos con más de una palabra
   {
```

Para entrar en esta parte del código e interpretar los comandos introducidos por el jugador relacionados con el pirata, deberemos comprobar que el pirata y el jugador se encuentran en la misma habitación. En caso de que no sea así, no se entrará en esta parte del código y se mostrará el típico mensaje de error cuando se introduce un comando incorrecto. De esta forma, si el jugador introdujera alguno de los comandos relacionados con el pirata sin haber descubierto previamente su existencia, no se le dará ninguna pista sobre ello.

Haciendo uso de *strtok* (cuya explicación se realizó en la entrega anterior), comprobamos si la primera palabra del comando del jugador es *decir*. En caso de que sea así, tratamos de analizar qué es lo que ha dicho el jugador. Usando *strtok* de nuevo extraemos el resto del comando introducido del jugador, y lo comparamos con la cadena "ayuda" (haciendo uso de la secuencia de escape \" para poder introducir comillas dentro de la cadena). En caso de que el jugador haya dicho exactamente eso, el pirata le dará la pista que necesita. En caso contrario, el pirata ignorará al jugador.

Como se puede comprobar, el pirata no es un hombre de muchas palabras. De todas formas, con esto quedan sentadas las bases para una posible ampliación de su vocabulario (lo cual se deja como ejercicio al lector).

FINALIZANDO NUESTRA AVENTURA

Por último vamos a permitir que Guybrush llegue a ser pirata. Para ello, vamos a introducir las modificaciones necesarias en el intérprete de comandos para que se pueda finalizar la aventura.

Como le dice el pirata a Guybrush si éste le pide ayuda, para poder llegar a ser un pirata de verdad deberá vencerle con la espada. Pero no es tan fácil, pues el pirata es un hábil espadachín que sólo podrá ser vencido si está ebrio. Por lo tanto, para finalizar la aventura, Guybrush deberá darle al pirata la jarra de grog que lleva desde el inicio de la aventura para que éste se emborrache, y luchar o pelear con la espada.

El código que debemos introducir en el intérprete de comandos, y dentro de la parte correspondiente a los comandos relacionados con el pirata, es el siguiente:

```
else if (localizacionPirata == habitacion)
// comandos del pirata
{
  if (!strcmp(comando,"dar jarra a pirata"))
  {
    if (objetos[1].localizacion == -1)
    {
      printf("\n\nEl pirata coge la jarra y se bebe su
contenido...\n\n");
      sleep(1);
      printf("\n\n... el pirata sigue bebiendo...\n\n");
      sleep(1);
      printf("\n\n... el pirata tiene sintomas evidentes de
embriaguez.\n\n");
      borrachoPirata = 1;
      objetos[1].localizacion = -2;
      pesoTransportado -= objetos[1].peso;
```

```
    }
  }
  else if (!strcmp(comando,"luchar con pirata") || !strcmp
(comando,"pelear con pirata"))
    {
      if (objetos[0].localizacion == -1)
      {
        if (borrachoPirata == 0)
        {
          printf("\n\nEl pirata te vence sin
dificultades...\n\n");
        }
        else
        {
          final = 1;
          printf("\n\nEl pirata esta tan borracho que no
puede
            ni aguantar la espada... \n\n");
          sleep(1);
          printf("\n\nTras un larga lucha...\n\n");
          sleep(1);
          printf("\n\nUna larga, larga, lucha...\n\n");
          sleep(1);
          printf("\n\nConsigues vencerle!!\n\n");
          sleep(2);
          printf("\n\n\n\nFELICIDADES. Has conseguido
completar la
            aventura y convertirte en un GRAN pirata!\n\n");
        }
     }
    }
    else
    {
      strcpy(palabra,strtok(comando," "));
      if (strcmp(comando,"decir") == 0)
      {
        strcpy(palabra,strtok(0,"\0"));
```

Se ha añadido código para interpretar dos comandos: *dar jarra a pirata* y *luchar con pirata* (o *pelear con pirata* (Siempre sería deseable que el jugador pudiera utilizar sinónimos para las acciones que deba realizar). Vamos a analizar cada una de estas partes por separado.

Con respecto al código para dar la jarra al pirata:

```
if (!strcmp(comando,"dar jarra a pirata"))
{
  if (objetos[1].localizacion == -1)
  {
    printf("\n\nEl pirata coge la jarra y se bebe su
contenido...\n\n");
    sleep(1);
    printf("\n\n... el pirata sigue bebiendo...\n\n");
    sleep(1);
    printf("\n\n... el pirata tiene sintomas evidentes de
embriaguez.\n\n");
    borrachoPirata = 1;
    objetos[1].localizacion = -2;
    pesoTransportado -= objetos[1].peso;
  }
}
```

Comenzamos por comprobar si el jugador tiene la jarra en el inventario (que el objeto 1, la jarra, tiene -1 como valor de localización, lo cual quiere decir que está en el inventario del jugador). En el caso de que sea así, mostramos tres mensajes de texto con printf informando al jugador de la nueva situación ebria del pirata. Entre mensaje y mensaje, para añadir mayor dramatismo, se hace uso de la función sleep (que forma parte de stdlib.h). El parámetro de sleep indica el número de segundos que se detendrá la aplicación. Lo siguiente es darle el valor de 1 a la variable borrachoPirata, almacenando el estado de embriaguez del mismo. Por último, hacemos que la localización de la jarra sea -2 (con lo cual no está en el inventario del jugador, y tampoco en ninguna de las habitaciones: es imposible que el jugador vuelva a encontrase con la jarra) y le restamos su peso al peso transportado por el jugador. La jarra está 'destruida' y no podrá volver a utilizarse en la aventura.

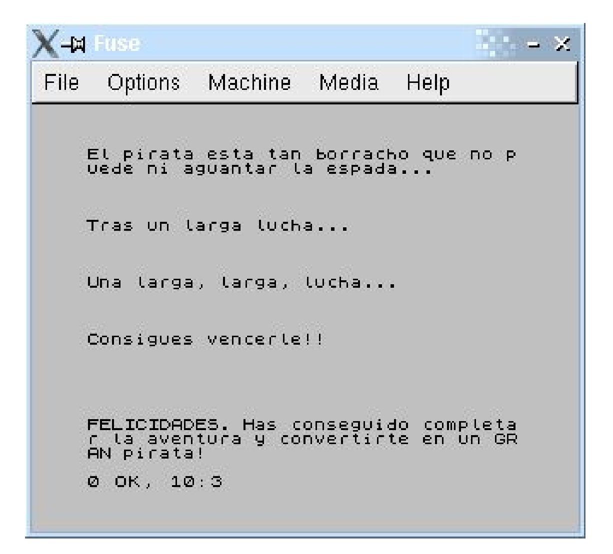
Y con respecto al código de la espada:

```
else if (!strcmp(comando,"luchar con pirata") || !strcmp
(comando,"pelear con pirata"))
{
  if (objetos[0].localizacion == -1)
  {
    if (borrachoPirata == 0)
    {
      printf("\n\nEl pirata te vence sin
dificultades...\n\n");
    }
    else
    {
      final = 1;
      printf("\n\nEl pirata esta tan borracho que no puede ni
        aguantar la espada... \n\n");
      sleep(1);
      printf("\n\nTras un larga lucha...\n\n");
      sleep(1);
      printf("\n\nUna larga, larga, lucha...\n\n");
      sleep(1);
      printf("\n\nConsigues vencerle!!\n\n");
      sleep(2);
      printf("\n\n\n\nFELICIDADES. Has conseguido completar
la aventura
        y convertirte en un GRAN pirata!\n\n");
    }
  }
}
```

Primero comprobamos si la espada está en el inventario del jugador (si el valor de localización del objeto 0, correspondiente a la espada, es -1). En el caso de que sea así, se comprueba si el pirata está borracho, según el valor de la variable borrachoPirata. Si vale 0, el pirata no está borracho, por lo que mostramos simplemente un mensaje de texto indicando que al estar sobrio el pirata vence al jugador sin dificultades.

En el caso contrario, la variable valdrá 1 y el pirata estará borracho. Le damos a la variable final el valor 1, indicando que el juego ha terminado. Si recordamos el código, veremos que al principio del intérprete de comandos había una sentencia while que hacía ejecutarse este intérprete hasta que el valor de la variable final cambiara. Pues bien, por fin podremos terminar la avenura. Lo siguiente es mostrar unos cuantos mensajes con printf, enfatizando con sleep, relatando la tremenda lucha con el pirata. Para el último mensaje, el de

felicitación, se deja una espera más larga, para añadir algo de tensión escénica.



No había presupuesto para un final más espectacular

¡Y ya está! Tenemos una aventura simple pero completa, que puede ser terminada por un jugador avispado.

RESUMEN

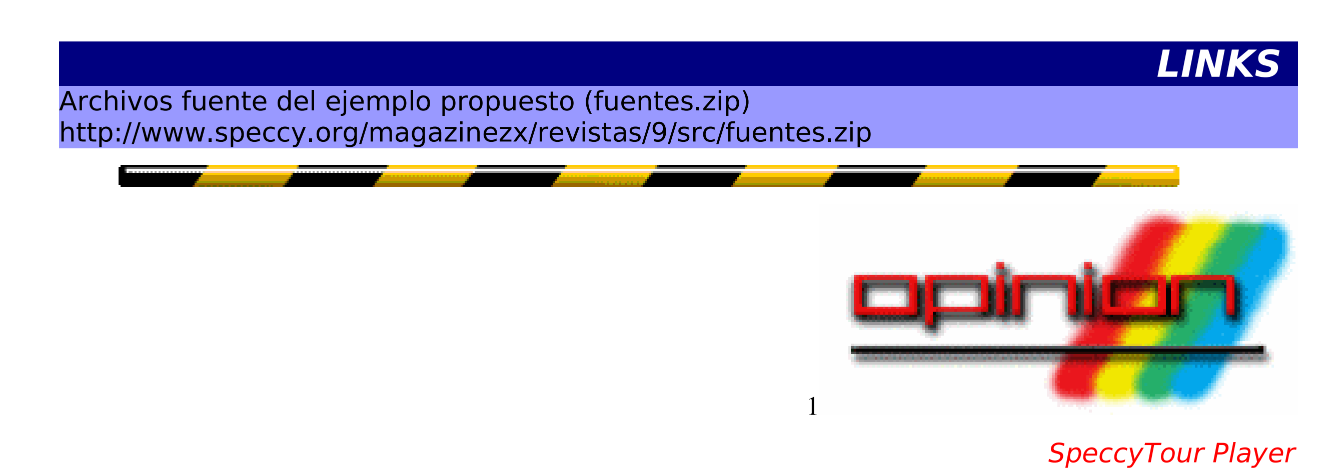
Durante estos tres artículos hemos desarrollado una sencilla aventura conversacional, con el objetivo de mostrar las características de las herramientas de texto de z88dk. Los resultados son poco espectaculares, pero hemos conseguido meter la cabeza y perder el miedo.

Lo que se habrá podido comprobar es que estas herramientas de texto son lentas. Esto quiere decir que solo se podrían utilizar con aplicaciones o juegos exclusivamente basados en texto. Para otro tipo de juegos, deberemos emplear otras técnicas, que iremos viendo conforme se presenten más entregas de z88dk.

¿Y a partir de ahora qué? Empezaremos a trabajar con conceptos gráficos y crear juegos más 'bonitos' visualmente. Veremos de qué capacidades hace gala z88dk para poder programar juegos de Spectrum en C. Pero todo esto, a partir del siguiente número...✍

LINKS

Archivos fuente del ejemplo propuesto (fuentes.zip)
http://www.speccy.org/magazinezx/revistas/9/src/fuentes.zip



1

SpeccyTour Player

SPECCY TOUR, un vistazo atrás y otro hacia delante.

Ya son cuatro las ediciones del Speccy Tour celebradas, desde que se crease el evento por Michael Bruhn en el año 2000. En esta primera competición los participantes mandaban snapshots de la puntuación como prueba de la puntuación obtenida. Era un sistema mediante el cual todo el mundo podía participar con el único requisito de una conexión a internet y un emulador. Como inconveniente, no había forma de controlar si un jugador salvaba y continuaba repetidamente su partida, dándose algunas suspicacias debido a la rapidez con la que algunos superaban sus records en los últimos compases de la competición. En líneas generales, esta competición fue un buen comienzo, y la competición se llevó con diligencia por parte de Michael.

En el 2001 Ramsoft hizo público su formato AIR, mediante el cual, el desarrollo de una partida podía grabarse y reproducirse tal y como se había producido. Bruhn apostó por esta tecnología sacrificando la accesibilidad de la competición. Se ganaba en seriedad y fiabilidad de resultados, pero se restringía la competición a los usuarios del citado emulador de Ramsoft, generando por ello no pocas críticas.

Con la perspectiva que proporciona el paso del tiempo, parece que acertó en esta apuesta, pues el éxito del formato produjo que la escena del Spectrum se pusiese manos a la obra para buscar alternativas a este formato, y a ser posible, que éste fuese universal, independientemente del emulador que se utilizase. En lo que al apartado organizativo se refiere, esta edición también se organizó de una forma correcta y ajustada a los plazos. El número de participantes creció a pesar de la restricción a usuarios de RealSpectrum, y de la cosiguiente limitación a usuarios del sistema operativo DOS.

En el 2002 cambiaron los miembros de la plantilla organizadora del Speccy Tour. Michael Bruhn cedió por decisión propia el testigo al Serbio Aleksandar Lukic y al Español Iván Ruiz Etxabe, que habían sido colaboradores en la edición anterior.

A estas alturas el RZX, el formato universal en lo que a grabación de partidas se refiere, ya había sido concebido, pero los emuladores que lo soportaban no tenían medidas de seguridad sobre el mismo, tales como el control de la velocidad de la emulación etc. Por tanto, una vez más, el RealSpectrum fue de inicio el único emulador de la competición. Claro está, la situación no era la misma del año anterior, puesto el número de usuarios de otras plataformas (linux) era creciente, y las quejas por ende, también. El criterio seguido por los nuevos directores era respetable, pero debieron ceñirse a él, y no introducir la posibilidad de usar RZX, sólo con el RealSpectrum para WIN32 (en una versión hecha para la ocasión) una vez iniciada ya la competición. Fue una decisión precipitada, porque no hubo tiempo material de hacer las pruebas pertinentes, y sólo solucionaba el problema a un reducido número de usuarios. La competición como tal, tuvo un incremento considerable de participantes, lo cual fue positivo. La competición se desarrolló normalmente hasta las últimas semanas en las que hubo un retraso considerable para publicar las tablas semanales.

Y por último el 2003. Mismos organizadores, con la añadidura de Kak.

En esta última edición los objetivos eran más ambiciosos. Nuevas plataformas, nuevos emuladores (Spectaculator, FUSE), jugadores de todo el planeta, premios...

Algunos de estos objetivos se salvaron dignamente, pero otros fueron un fracaso. La novedad de los nuevos emuladores trajo consigo unos problemas de compatibilidad inicialmente, aunque se puede entender que no se hiciesen las pruebas oportunas con debida antelación, ya que se logró la inclusión de los mismos a última hora, y la demanda de estos era grande.

El aspecto más criticable fue una vez más cómo se gestionaron las últimas semanas de tour, la publicación de los resultados definitivos se alargó no semanas, sino meses, mientras la página del tour decía que iba a ser inminente, ¿será el influjo de RAMSOFT?.

Lo malo de este tipo de situaciones es que desaniman al jugador, si se pone mucho énfasis en la seguridad del formato y en que cumpla una serie de requisitos, es porque la competición quiere tener tintes serios, y por tanto su gestión debe estar a la altura de las circunstancias.

No fue así. Uno se podría preguntar para qué se pone tanto énfasis en que los emuladores tengan medidas de seguridad si luego están meses sin confirmar las partidas. Se estaría dando la razón a los que opinan que la competición habría de ser más relajada.

Otro de los aspectos que dejó la sensación de improvisar es la selección de juegos. Tanto en la última como en la anterior edición hubo polémicas con algún juego.

En el caso al que me refiero, el Arcadia, a las tres semanas ya había jugadores ciclando indefinididamente el juego. Esto ya había ocurrido, de otra guisa, con el Raid over Moscow la edición anterior. Bien es verdad que son los propios jugadores los que eligieron los juegos a jugar, pero el pozo de juegos desde donde se escoge debería haber sido más estudiado, sin lugar a dudas. Maziacs fue otro ejemplo de que este problema no es un hecho puntual. Además no hay que olvidar que tal y como fueron montadas las cosas, cualquiera podía registrarse n veces, aduciendo pertenecer a las Islas Salomon o Krakatoa y votando por su juego favorito. Es a todas luces claro que al fin y a la postre sólo el 38% de los apuntados luego realmente disputa la competición, por lo que la justicia del sistema actual de votaciones queda, cuando menos, en duda.

Sin embargo, de lo que no cabe duda es que tanto Ivan, Aleksandar, Kak... hacen esto porque les gusta, y que nos proporcionan buenas dosis de diversión.

Esperemos que con las cosas más asentadas y con menos inminencias las cosas vuelvan a su cauce en el 2004. Ojalá, pues esta competición sólo tiene un equivalente en máquinas recreativas con el MAME, y es algo de lo que no pueden cacarear los usuarios de otras plataformas. ✍



HORACE

Entrevista a Droy

Si hay un nombre que destaque en el panorma estatal (y, seguramente, en el internacional) en lo que concierne a desarrollo de nuevo hardware para nuestros Spectrum, éste sería el nombre de nuestro entrevistado. Sus "trastos" permiten ampliar y añadir nuevas funcionalidades al, ya de por sí, limitado diseño original del Spectrum.

"Me proporciona mucho placer retomar mis sueños de chaval y llevarlos a cabo."



Una pregunta inevitable para empezar esta entrevista: ¿cómo te iniciaste en el mundo del Spectrum y , sobre todo, en el ámbito de la electrónica?

Se podría decir que mi afición por la electrónica empezó cuando mis padres me regalaron un Electro-L (un juego de montajes eléctricos). Años despues llegó el Scatron (juego de montajes electrónicos) y desde entonces no he parado de disfrutar aprendiendo electrónica, reparando y diseñando.

A comienzos de los 80 disponía de una subscripción a una revista llamada "Revista Española de Electrónica", fue un regalo de mis padres por algo que ya no recuerdo. Esta revista disponía, además de las secciones tipicas de montajes y divulgación, de una sección llamada "Micro-bit". En esta sección se publicaban programas para el ZX81, Jupiter, etc. Con la revista en la mano unos amigos y yo nos acercamos a una tienda de electrodomésticos donde vendían ZX81, y fue alli donde tomé contacto por primera vez con un ordenador personal, tecleando programas de "Micro-bit". Al principio al pie de la letra, pronto realizando pequeñas modificaciones, luego creando programas completos.

La cosa fue bien para la tienda en cuestión, que nos dejaba usar el ZX81 de exposición, ya que ayudábamos a vender muchos ordenadores. Supongo que los padres indecisos nos veían programando y querían algo así para sus hijos.

Tras convencer a mis padres para que pagasen 14725 pts. por un ZX81 de 1Kb tuve la "suerte" de que no quedaban. Un vendedor hábil y mi hermano mayor convencieron a mi padre para que me comprase el nuevo modelo ZX Spectrum 16K a todo color (más de 49000 pts.). A partir de ese momento el Spectrum pasó a formar parte de mi vida. Pronto aprendí a programar en código máquina, recuerdo perfectamente el momento en el que comprendí como el procesador trabajaba y que el "editor del basic" era solo una capa, un programa en ROM para facilitar las cosas, pero que el código máquina siempre estaba allí. Es uno de mis recuerdos más bonitos del Spectrum, pues fue como una revelación.

¿Cómo te entró el gusanillo de la creación de nuevo hardware para el Spectrum?

Cuando era un chaval con mi Spectrum nuevo no dejaba de leer revistas de electrónica e informática (la ZX era mi favorita, luegó llego la Microhobby) y en todas se anunciaba hardware que para mi era inalcanzable económicamente (ZX-Interface I, Disciple, etc). De alguna forma esto quedó como una asignatura pendiente, pues tenía ideas pero no medios ni tecnología. Ahora que dispongo de más medios que cuando tenía 14 años, y más conocimientos y tecnología, me proporciona mucho placer retomar mis sueños de chaval y llevarlos a cabo.

Después conocí a Jose Manuel, el webmaster de El Trastero (http://www.speccy.org/trastero/) y al enterarse de que fabricaba cartuchos para el Interface II me animó a poner en su página algo al respecto, primero fueron unas fotos, luego algo de información, poco a poco me fue animando hasta lo que es ahora. Así nació la sección "Los Trastos de Droy".

¿La actividad que desarrollas profesionalmente tiene que ver con este hobby? ¿O es un hobby aprendido por tu cuenta?

No, aunque ahora trabajo para una multinacional de la informática, mi actividad poco tiene que ver con este

hobby. Precisamente por eso disfruto más aún de esta afición.

Yo estudié Maestría industrial (los cinco años de FP electrónica) donde se obtiene una buena base, pero es sólo eso, una base. Lo que sé ahora es el resultado de muchos años de leer libros, revistas especializadas, experimentar mucho, en definitiva de cultivar una afición.

¿Qué herramientas y materiales consideras indispensables para cualquier usuario que quiera realizar alguno de tus diseños?



Imagen del taller de trabajo de Droy

Lo único realmente indispensable es sólo el soldador y los componentes de cada montaje ;-) En realidad lo que hace realmente mucha falta es paciencia, ganas e ilusión, pero, sobre todo, paciencia.

Existen varios proyectos que has realizado y que están disponibles en El Trastero del Spectrum (http://www.speccy.org/trastero/cosas/droy/droy.htm), como por ejemplo el Interface II y sus cartuchos: ¡no me puedo creer que sea tan aparentemente sencillo "clonarlos"! ¿Dónde está el truco? :-)

La verdad es que los cartuchos fue lo primero que publiqué. Me sorprendió mucho al ver el esquema original del Interface II que lo relacionado con el cartucho no tenía electrónica, que sólo eran conexiones del bus hacia el cartucho (la parte del joystick es otra cosa...). Entonces intenté imaginarme como debia ser el cartucho para que un adaptador así funcionase, no fue complicado. Pero había algo que debía confirmar sobre un cartucho original (si la señal de ROMCS era anulada por el propio cartucho). Escribí por internet a alguien que poseía uno y le pedí que midiese en dos contactos de un cartucho original. La respuesta confirmó mi sospecha, el cartucho ya estaba diseñado. El Interface II original, como ya sabes, no son más que conexiones, solo había que aislarlas de la parte del joystick.

Los cartuchos de 32K son el resultado de un momento de inspiración, es algo que disfruté mucho haciendo. Cuando por fin cargó el Knight Lore en 3 seg. desde cartucho me acordé de lo que deseaba esto cuando tenía 14 años y fue muy gratificante.

Vistas las mejoras y modificaciones que has hecho con el Jupiter Ace, ¿te atreverías a hacer algo parecido con un clon del Spectrum al estilo del Sprinter, SpeccyBob o Chrome?

Estos clones estan muy bien diseñados, en especial el Sprinter que es un producto muy profesional y muy bien acabado, difícil de superar. No veo que necesiten mejoras, o al menos, no veo qué pueda mejorar yo. En este momento me atrae más el ampliar el Spectrum con nuevos adaptadores.

¿Cuál de todos tus diseños entrañó mayor dificultad en su desarrollo? Y, por contra, para los novatos, ¿cuál nos aconsejas realizar por su sencillez?

Sin duda el IMAR-3 fue el más complicado pero a su vez con el que más disfruté. El diseño y desarrollo de este adaptador me hizo disfrutar mucho, y aprendí un montón, pues no se trataba de rediseñar o modificar sino de hacer algo partiendo de cero. Es del que me siento más orgulloso. Creo que desde el punto de vista de aficionado no es de los complicados de llevar a la práctica. Pero para simplicidad, el Interface II y los cartuchos.

A la hora de hacer estos diseños de hardware para ordenadores de hace más de 20 años, te habrás encontrado con problemas a la hora de encontrar ciertos componentes. ¿Se siguen fabricando aún muchos de ellos? ¿Te resulta complicado encontrar estos componentes o sustitutos para ellos?

La verdad es que en el caso del PlusD tuve algunas dificultades para encontrar el WD1772. Este integrado no se fabrica desde hace años, y lo peor es que realmente fue diseñado inicialmente por encargo de Atari. Una vez que Atari finaliza la producción, se acabó la distribución. Al final lo traje de EEUU y fue una cuestión de algo más de dinero que si lo compras en la tienda de la esquina. En el caso del Jupiter la sustitución de componentes antiguos por actuales fue más un reto que una dificultad. Además, si traía los componentes de EEUU, o los sacaba de una placa antigua, yo tendría mi Jupiter, pero no facilitaba nada a nadie para que pudiera hacer el suyo.

Algo que me motiva mucho es que las cosas sean realizables por los demás sin tener que pedir componentes por internet. Este fue un problema que me tuve que plantear cuando diseñe el IMAR-3: habría sido infinitamente más facil de diseñar si hubiera incluido un cross-switch, y 100% compatible, seguramente el adaptador de Sintech lleve uno. Pero eso dejaba fuera al 90% de los usuarios, pues estos integrados no los encuentras así como así. Acepté el reto y lo diseñé usando componentes comunes, aunque esto le hace perder alguna funcionalidad.

¿Existe la posibilidad de conseguir estos diseños tuyos ya fabricados? Y si no es así, ¿se ha pensado en realizar alguna distribución a pequeña escala de alguna de tus creaciones?

Verás, este es un punto por el que mucha gente se interesa y me pregunta. Yo siempre respondo lo mismo: no tengo interés en sacar beneficio económico de ninguna de mis creaciones pues eso requeriría mucho tiempo del que no ando muy sobrado. Para mí es un hobby y quiero que siga siéndolo. Lo único a lo que sí estoy dispuesto es a intercambiar puntualmente algun adaptador hecho por mi

por alguna pieza de hardware del Spectrum o similar.

Por otro lado si alguien quiere distribuir a pequeña escala alguno de mis diseños no tengo inconveniente, ni voy a pedirle comisión o derechos de autor.

No sólo te dedicas a fabricar los montajes, sino que también te ocupas de su creación, desarrollo y posterior documentación. ¿Cuál de estas partes requiere de más tiempo y de más trabajo por tu parte? ¿Y cuál de ellas es la que menos te gusta?

La documentación requiere más tiempo del que yo imaginaba, y es sin duda lo que menos me gusta. Pero en cierta forma me ayuda a mantener la disciplina el pensar que lo que estoy haciendo lo tendré que documentar, así que me fuerza a llevar mayor control y registro de lo que hago, en otras palabras, trabajas mejor.

¿En qué nuevos proyectos andas inmerso últimamente? ¿Nos puedes adelantar algo en exclusiva?

Tengo varias cosas en la cabeza. Una es un adaptador de disquete con componentes "encontrables" (no como el WD1772 del PlusD) y que lea snaps grabados en disquetes de 1.44 Mb con un PC directamente (*.SNA *.Z80). Algo sencillo y fácil de hacer y que nos permita cargar snaps en un ZX Spectrum 48K real.

Otra idea es un sistema de cartuchos regrabales desde el PC con capacidad de 512 Kb (unos 10 snaps *.SNA por cartucho).

Y aún ando dándole vueltas al adaptador ISA para Spectrum. Este proyecto está aparcado pero no olvidado. Para los que no lo conozcan os resumo que es algo que empecé hace tiempo, se trata de un dar al Spectrum una ranura ISA para poder usar los adaptadores multifunción (FDC / IDE / Serie / Centronics) y así poder tener disquetera reusando los adaptadores viejos de PCs ISA. Si finalmente pongo en marcha el primer proyecto, éste quedará aparcado definitivamente.



Droy con algunos años menos: corrían otros tiempos... para todos

Aparte de crear diseños nuevos, también has dedicado parte de tu tiempo a reparar Spectrums averiados, ¿cuál es el problema más habitual con el que te has encontrado? ¿Consideras que el Spectrum era un ordenador propenso a sufrir averías?

Los problemas más tipicos, aparte de la membrana del teclado, son los relacionados con la power supply. No me refiero al transformador externo, hablo de la parte del circuito del Spectrum que partiendo de 9V sin regular genera todas las tensiones necesarias (+9V regulados, +5V, +12V y -12V). Casi siempre es el transistor de oscilación.



Droy, en la actualidad, en su taller de diseño

La verdad es que las primeras issues tenían muchos problemas. La prueba de ello es la cantidad de ellas que hay. Y no hay que olvidar que la búsqueda de reducción de costes de fabricación no soló nos dejó un teclado de goma. De todas formas, teniendo en cuenta el precio que tenia en el mercado, no le podíamos pedir más a los chicos de Sinclair.

El esquema de diseño de la ULA del Spectrum es, aún hoy en día, un completo desconocido. ¿Crees que sería posible, sin contar con los esquemas, poder clonarla al 100%?

Bueno, en cierta forma Sprinter y Speccybob lo hacen. Por otro lado, el esquema "original" realizable con componentes comunes está disponible, lo que pasa es que cuando lo ves entiendes por que Sir Sinclar le encargo a la Ferranti que le fabricase un integrado a medida. Es más complejo que el resto del Spectrum.

Imaginemos que nos econtramos en los primeros años 80, que trabajas en Sinclair Research Ltd. y que Clive Sinclair llama a tu puerta para encargarte el diseño de lo que sería el nuevo ordenador ZX Spectrum. ¿Cómo hubiera sido tu diseño? ¿Que le hubieras añadido o cambiado respecto al diseño que finalmente se hizo?

Creo que en la mente de todos estaría un teclado mejor, algo como el del Oric Atmos, un bus de expansión más racional que facilite la vida a los diseñadores de hardware, un sistema de almacenamiento externo de alta velocidad...

Si lo piensas bien cualquier pequeño cambio en nuestro querido Spectrum podría dar al traste con el éxito que tuvo. Sinceramente no cambiaría nada, me encanta como fué, como es, en su simplicidad está también el secreto de su éxito.

En el caso del Spectrum 128k y sobre todo, en el caso del QL, ya es otra cosa, pero eso es otra historia.

Nada más, Droy, gracias por atendernos y esperamos que nos sigas deleitando con más creaciones tuyas para nuestro Spectrum.

Ha sido un placer, gracias a ti por ayudar a conservar la memoria de unas máquinas tan fantásticas y que nos evocan tantos buenos momentos. Un saludo para todos los aficionados al Spectrum. ✍